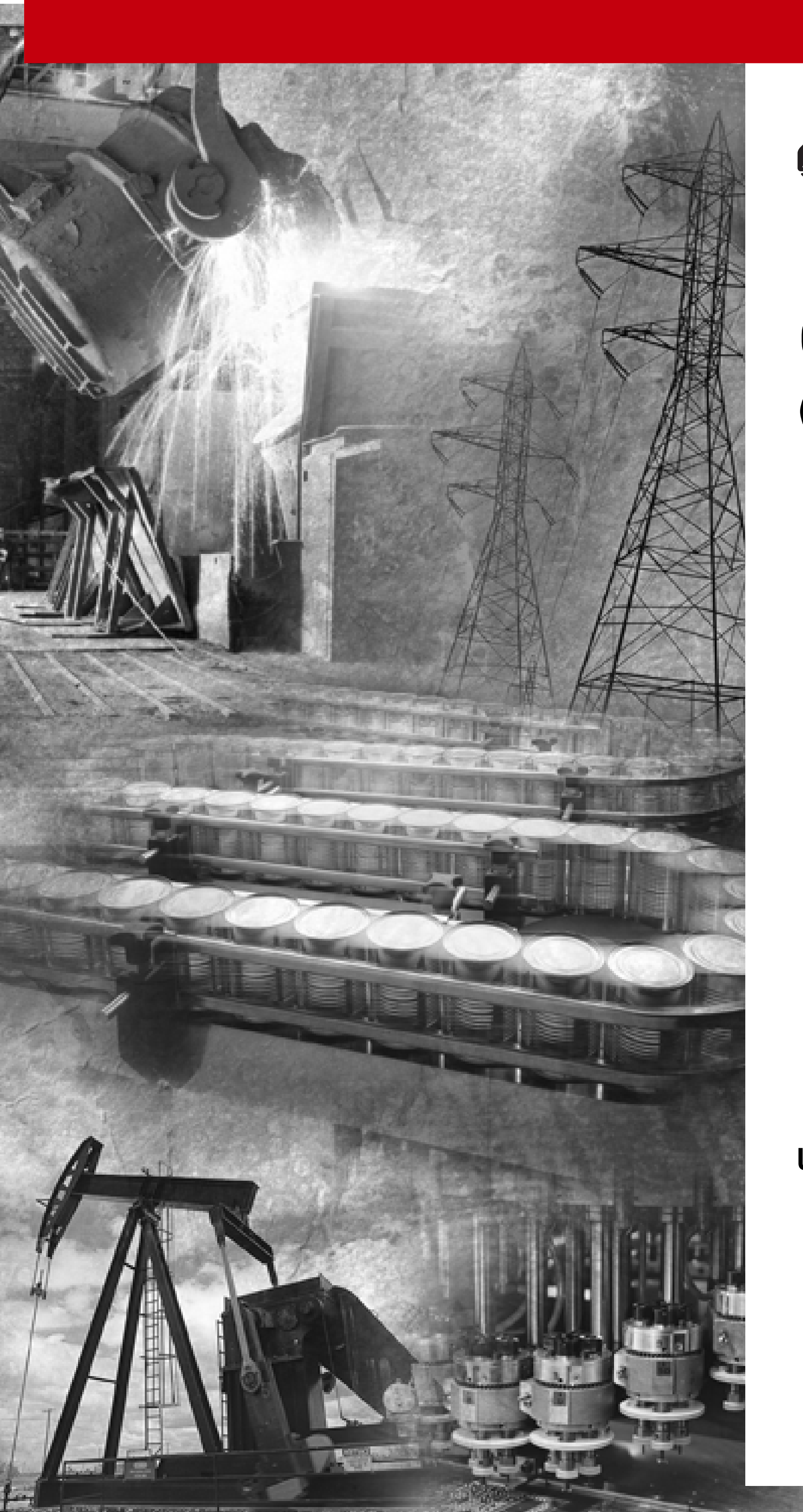


Allen-Bradley

CompactLogix System

(Catalog Number 1769-L20)

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION ⚠	Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss

Attention statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

Placing, Configuring, and Monitoring Local I/O

Chapter 3

Communicating with Devices on a Serial Link

Chapter 4

Chapter 5

Communicating with Devices on a DH-485 Link

Appendix A

CompactLogix System Specifications

Appendix B

CompactLogix Troubleshooting

Appendix C

Maintaining the Battery

Appendix D

Execution Time

Read this preface to familiarize yourself with the rest of the manual. This preface covers the following topics:

- who should use this manual
- how to use this manual
- related publications
- conventions used in this manual
- Rockwell Automation support

Who Should Use This Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use Allen-Bradley CompactLogix™ controllers.

How to Use This Manual

As much as possible, we organized this manual to explain, in a task-by-task manner, how to install, configure, program, operate and troubleshoot a CompactLogix control system.

Related Documentation

The table below provides a listing of publications that contain important information about CompactLogix systems.

For	Read this document	Document number
Information on installing the CompactLogix5320 controller	CompactLogix™ Modular Processor	1769-IN047A-EN-P
An technical overview of the CompactLogix system	CompactLogix™ Technical Data	1769-TD003A-EN-P
Information on common procedures using RSLogix 5000 software.	Logix5000 Controllers Common Procedures Programming Manual	1756-PM001B-EN-P
Indepth information on the CompactLogix Instruction Set	Logix5000 Controllers General Instruction Set Reference Manual	1756-RM003C-EN-P
Information on installing, configuring, and using Compact Analog I/O modules	Compact I/O Analog Modules User Manual	1769-UM002A-EN-P
Information on using the 1769-ADN DeviceNet Adapter.	Compact I/O 1769-ADN DeviceNet Adapter	1769-UM001A-US-P
In-depth information on grouding and wiring Allen-Bradley programmable controllers.	Allen-Bradley Programmable Controller Grounding and Wiring Guidelines	1770-4.1

If you would like a manual, you can:

- download a free electronic version from the internet at **www.theautomationbookstore.com**
- purchase a printed manual by:
  - contacting your local distributor or Rockwell Automation representative
  - visiting **www.theautomationbookstore.com** and placing your order
  - calling 1.800.963.9548 (USA/Canada) or 001.330.725.1574 (Outside USA/Canada)

Conventions Used in This Manual

The following conventions are used throughout this manual:

- Bulleted lists (like this one) provide information not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 Sales/Support Offices, 512 authorized distributors and 260 authorized Systems Integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreement

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, please review the information in Appendix B, CompactLogix Troubleshooting, first. Then call your local Rockwell Automation representative.

Your Questions or Comments on the Manual

If you find a problem with this manual, please notify us. If you have any suggestions for how this manual could be made more useful to you, please contact us at the address below:

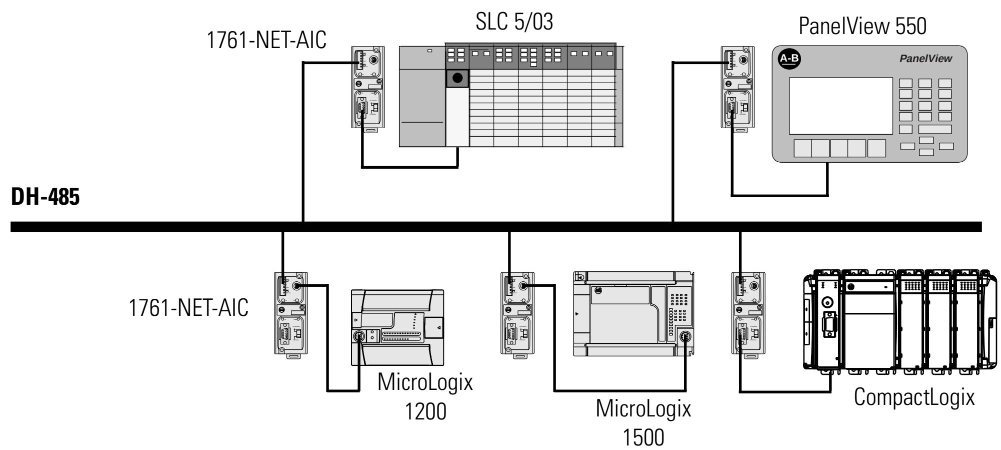
Rockwell Automation
Automation Control and Information Group
Technical Communication, Dept. A602V
P.O. Box 2086
Milwaukee, WI 53201-2086

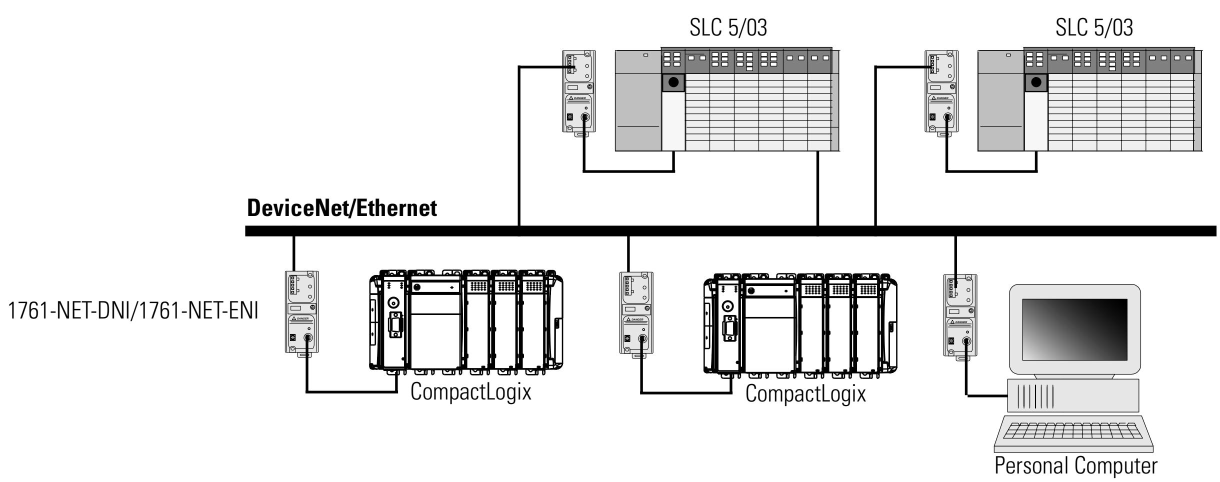
Getting Started

Introduction

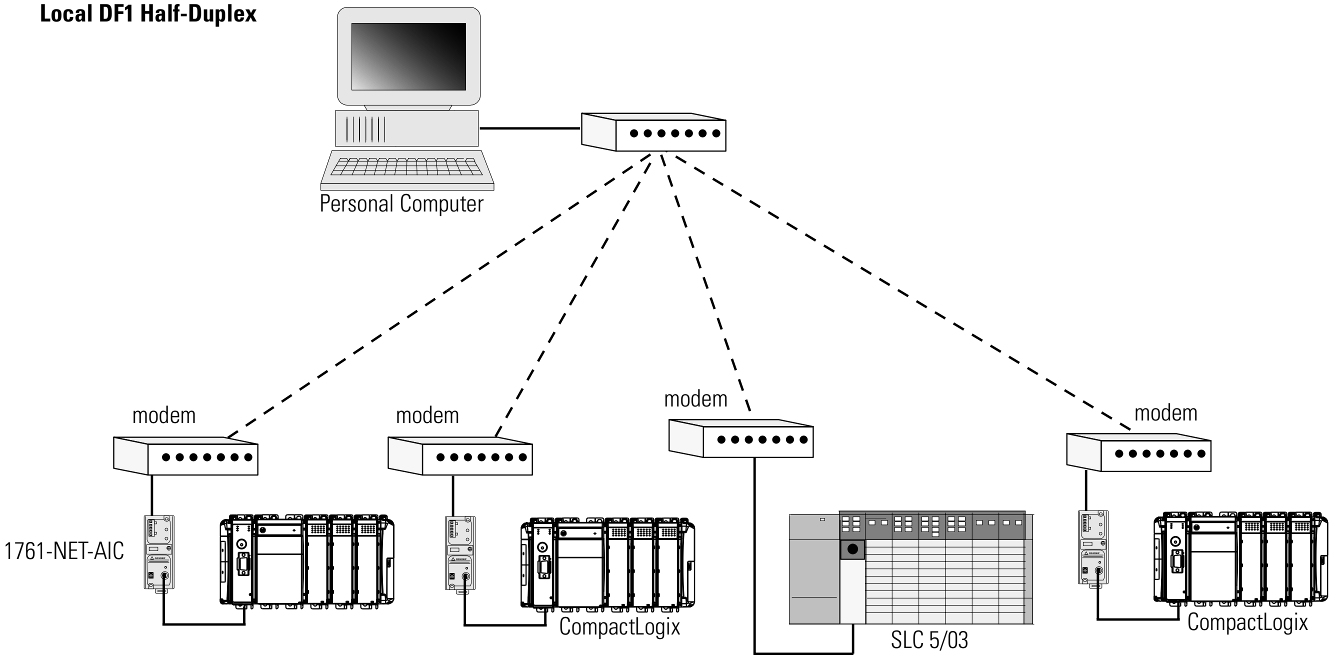
This chapter introduces the CompactLogix controller and provides a quick overview on creating and downloading a project. The steps in this chapter introduce the basic aspects of the CompactLogix controller.

The CompactLogix controller offers state-of-art control and I/O modules in a small cost-effective package.





Local DF1 Half-Duplex



The CompactLogix systems above illustrate some of the CompactLogix controller's capabilities. CompactLogix systems support the following:

- CompactLogix controllers at several locations communicating over DH-485, DeviceNet, or Ethernet via peer-to-peer messaging.
- RTU functionality using dial-up modems and DF1 Full-Duplex protocol.
- RTU functionality with radio frequency or leased-line modems and DF1 Half-Duplex protocol.
- CompactLogix controller controlling a maximum of 8 local Compact I/O modules.
- Local RS-232 connection for controller project upload/download, DF1 Full-Duplex communications, DH-485 networking, or for ASCII communications.
- Remote programming over DeviceNet and Ethernet.

Compliance to European Union Directives

This product has the CE mark and is approved for installation within the European Union and EEA regions. It has been designed and tested to meet the following directives.

EMC Directive

This product is tested to meet Council Directive 89/336/EEC Electromagnetic Compatibility (EMC) and the following standards, in whole or in part, documented in a technical construction file:

- EN 50081-2
  EMC - Generic Emission Standard, Part 2 - Industrial Environment
- EN 50082-2
  EMC - Generic Immunity Standard, Part 2 - Industrial Environment

This product is intended for use in an industrial environment.

Low Voltage Directive

This product is tested to meet Council Directive 73/23/EEC Low Voltage, by applying the safety requirements of EN 61131-2 Programmable Controllers, Part 2 - Equipment Requirements and Tests.

For specific information required by EN 61131-2, see the appropriate sections in this publication, as well as the following Allen-Bradley publications:

- *Industrial Automation Wiring and Grounding Guidelines for Noise Immunity*, publication 1770-4.1
- *Guidelines for Handling Lithium Batteries*, publication AG-5.4
- *Automation Systems Catalog*, publication B113

Safety Considerations

Safety considerations are an important element of proper system installation. Actively thinking about the safety of yourself and others, as well as the condition of your equipment, is of primary importance. We recommend reviewing the following safety considerations.

Hazardous Location Considerations

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, D or non-hazardous locations only. The following WARNING statement applies to use in hazardous locations.

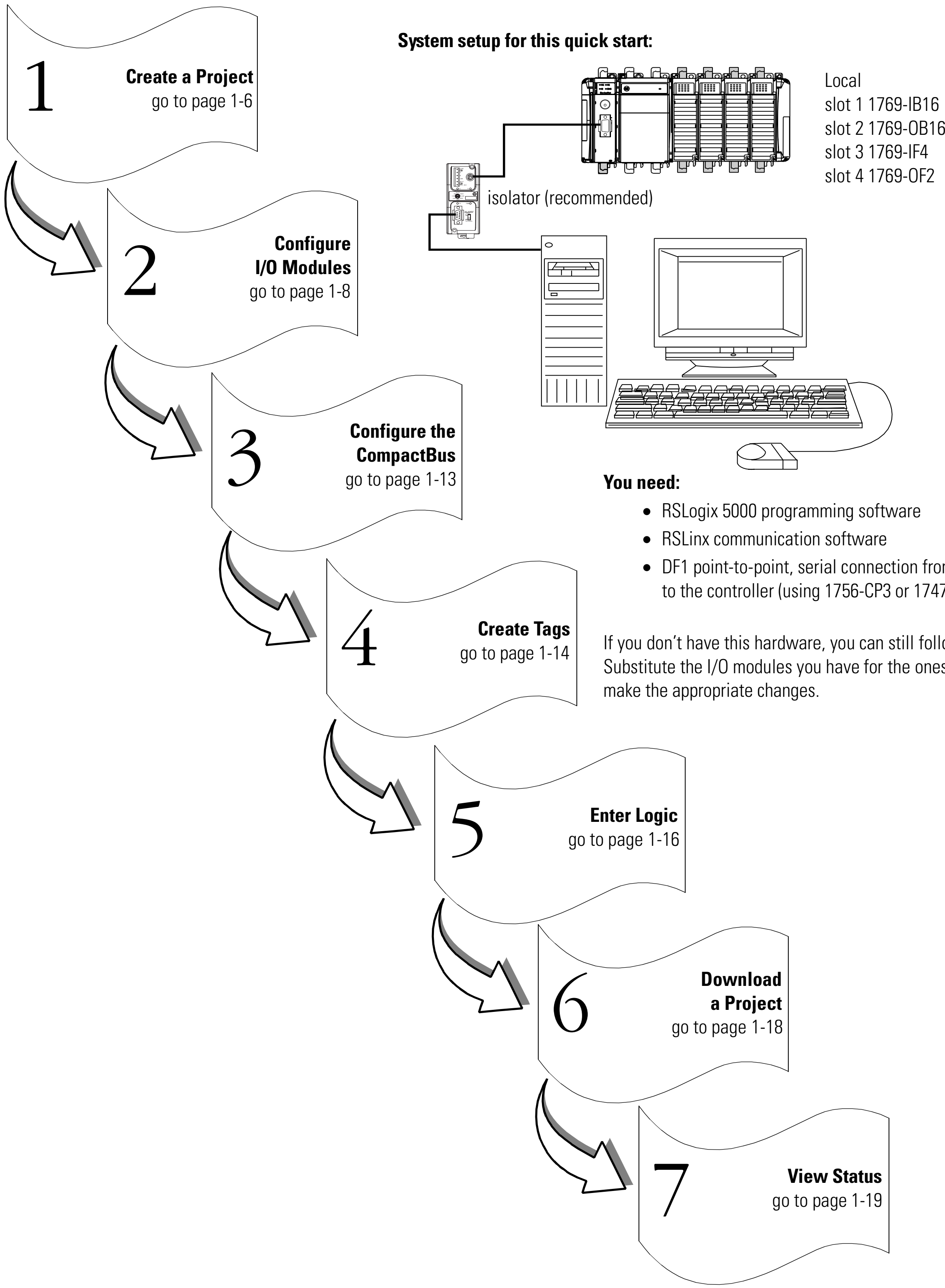
WARNING

EXPLOSION HAZARD

- Substitution of components may impair suitability for Class I, Division 2.
- Do not replace components or disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
- Do not connect or disconnect components unless power has been switched off or the area is known to be non-hazardous.
- This product must be installed in an enclosure. All cables connected to the product must remain in the enclosure or be protected by conduit or other means.
- All wiring must comply with N.E.C. article 501-4(b).

Creating and Downloading a Project

The following diagram illustrates the steps you follow to create and download a project. The remainder of this chapter provides examples of each step.



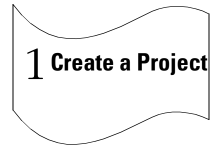
System setup for this quick start:

You need:

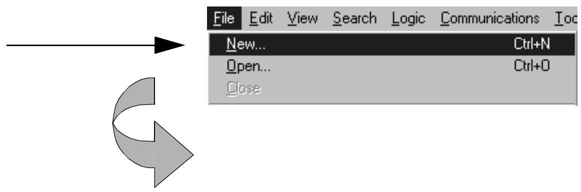
- RSLogix 5000 programming software
- RSLinx communication software
- DF1 point-to-point, serial connection from the workstation to the controller (using 1756-CP3 or 1747-CP3 cable)

If you don't have this hardware, you can still follow these steps. Substitute the I/O modules you have for the ones listed here and make the appropriate changes.

Creating a Project



1. **Select File → New.**



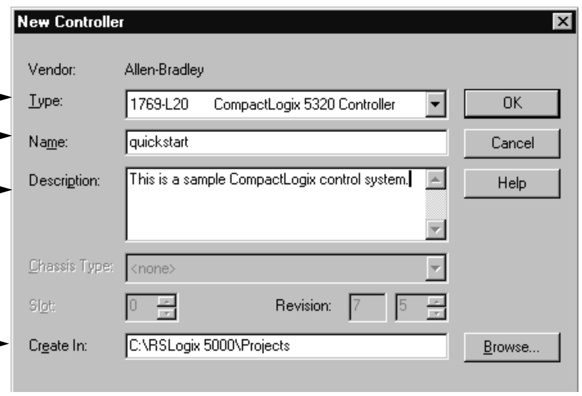
2. **Define the project.** The software uses the project name you enter with an .ACD extension to store your project.

Select a controller type.

Name the project.

Describe the project (optional).

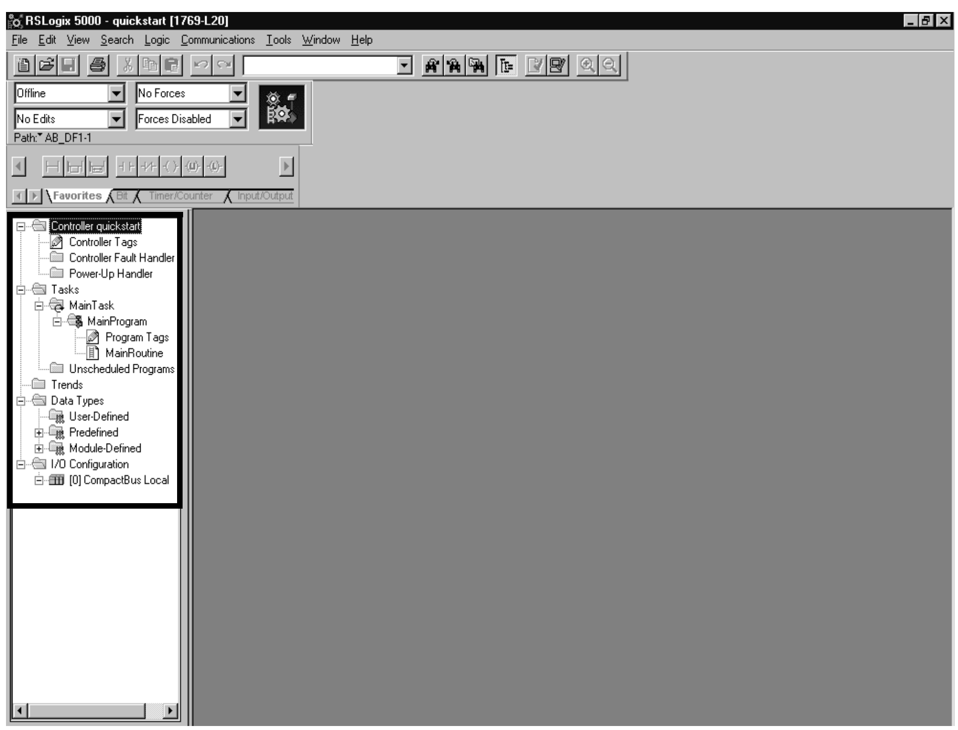
Select where to store the project (typically use the default directory).



Click OK.

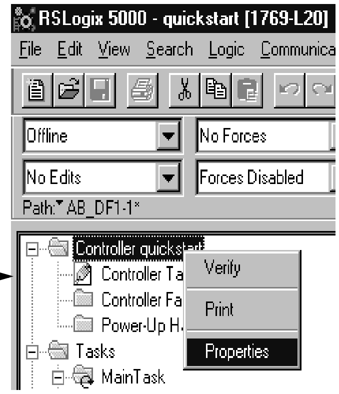
The software creates the new project and displays:

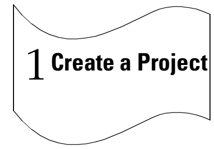
controller organizer



Changing Project Properties

1. **View properties for Controller quickstart.**

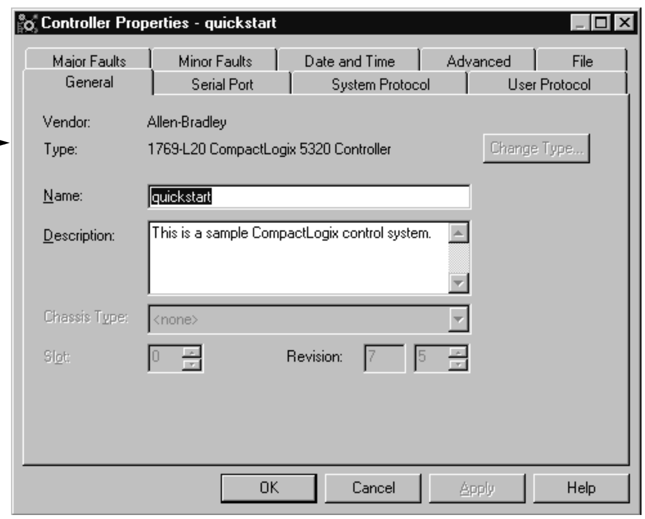




   A. Place the cursor over the Controller quickstart folder.

   B. Click the right mouse button and select Properties.

2. **View the General tab.**
   The screen defaults to the General tab.



Verify that the controller settings are correct. Make changes if necessary.

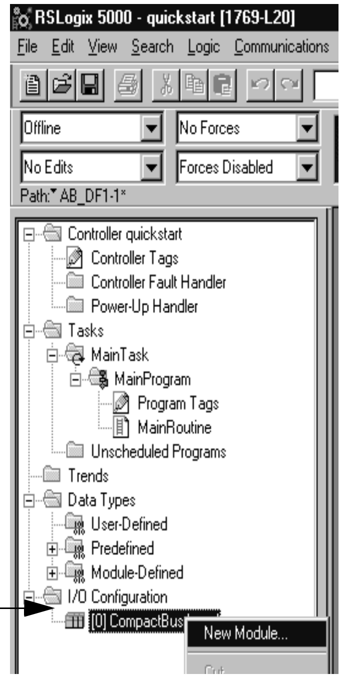
Click OK.

2 **Configure I/O Modules**

Adding a Local I/O Module

1. **Create a new module.**

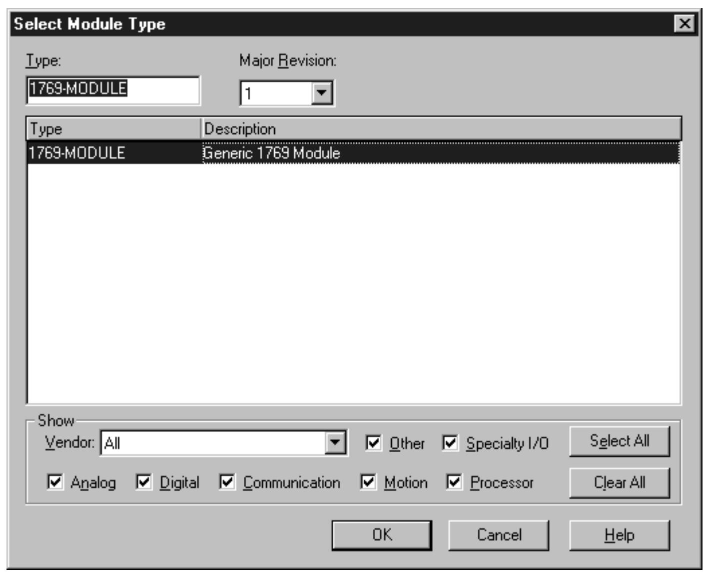
NOTE You need to add a local I/O module to the project for every module connected to your controller.



A. Place the cursor over the [0] Compact Bus Local.

B. Click the right mouse button and select New Module.

2. **Select the Generic module type.**



Click OK.

continued

Adding a Local I/O Module (*continued*)

2 **Configure I/O Modules**

3. Identify the module.

These screens are specific to the Generic module.

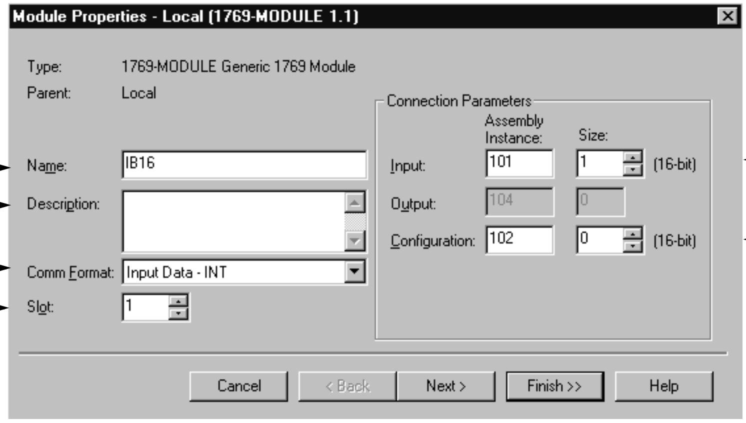
You should enter a name.

Describe the module (optional).

Select the communication format.

Specify slot number.

Click Next.



Enter Input Image Size.

Enter Configuration Image Size.

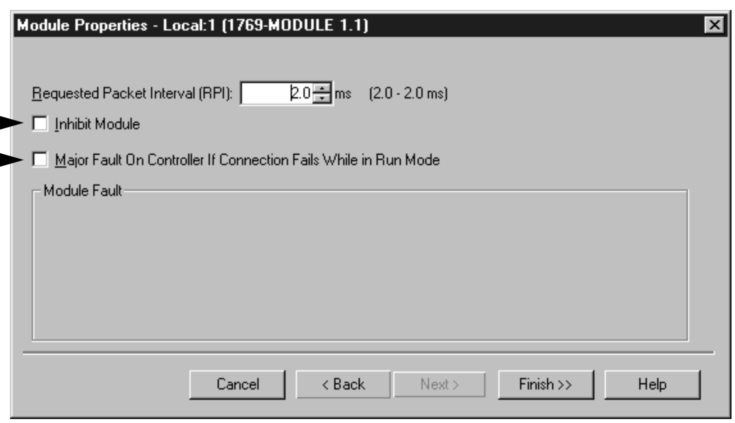
NOTE The above example shows the settings for 1769-IB16 in slot 1. The accessible fields change depending upon the type of module and the Comm Format chosen. Discrete input modules are not configurable. For additional information (comm format, assembly instances, and sizes) and 1769 generic module application examples, see Appendix G, *Configuring I/O with the 1769 Generic Profile*.

4. Show "Next" screen.

Using this screen, you can inhibit or uninhibit each module.

You can also configure modules to generate a major fault in the controller if they lose their connection to the controller.

See *Configuring the Module's Response to a Connection Failure* on page 3-11 and *Inhibiting I/O Module Operation* on page 3-9 for more information on these functions.



Click Next.

Click Finish.

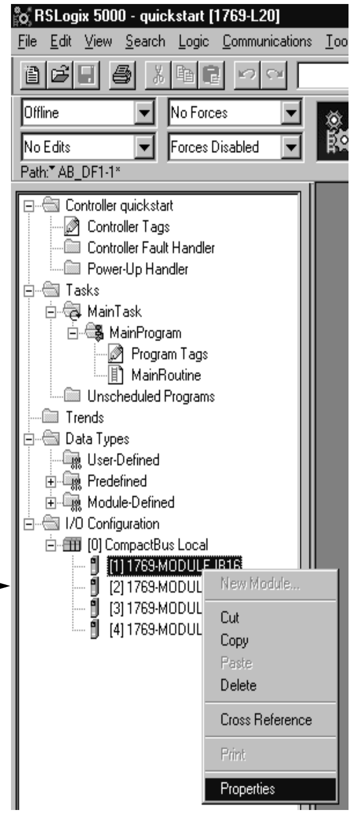
5. Repeat steps 1 through 4 for each I/O module in your system.

NOTE If you are using hold last state or user-defined safe state data for outputs, enter configuration data in tag base. See Appendix G, *Configuring I/O with the 1769 Generic Profile*, for additional information and generic profile application examples.

Changing Module Properties

2 Configure I/O Modules

1. **View properties for the module.**



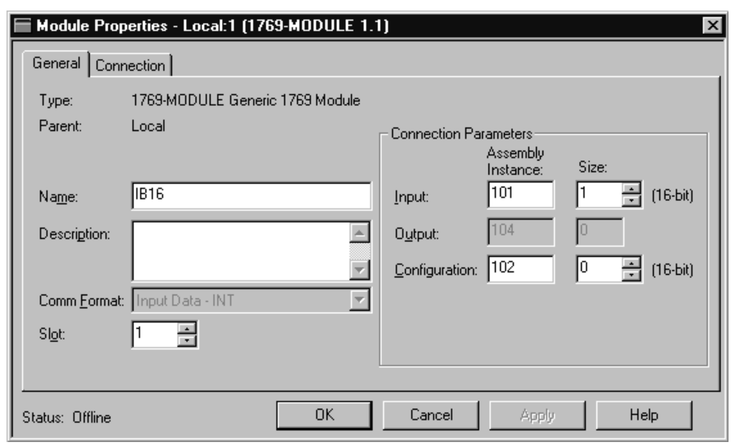
A. Place the cursor over the 1769-IB16 module.

B. Click the right mouse button and select Properties.

2. **View the General tab.**
   The screen defaults to the General tab.

Verify that the module settings are correct. Make changes if necessary.

Click OK.



The parameters that appear on this General tab depend on the type of module.

IMPORTANT	If you want to change the Comm Format of a module, you must first delete the module and then re-add it using the communication format you want.

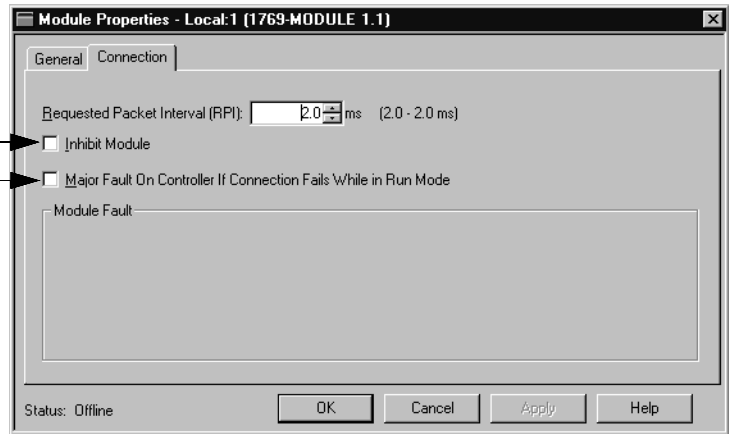
For information and application examples on generic profiles, see Appendix G, *Configuring I/O with the 1769 Generic Profile*.

3. View the Connections tab.

Using this screen, you can inhibit or uninhibit each module.

You can also configure modules to generate a major fault in the controller if they lose their connection to the controller.

See *Configuring the Module's Response to a Connection Failure* on page 3-11 and *Inhibiting I/O Module Operation* on page 3-9 for more information on these functions.



Viewing I/O Tags

2 **Configure I/O Modules**

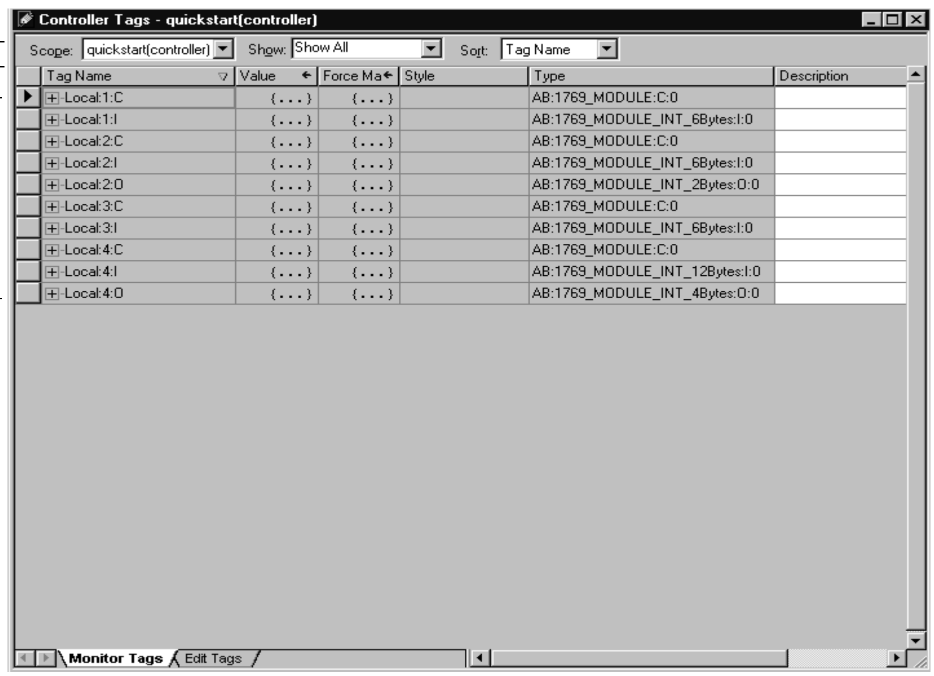
1. **View the tags for the controller.**

Place the cursor on the Controller Tags folder and double-click.

The software displays the module-defined tags for the I/O modules you created.

Tag Scope(1)

1769 module tags



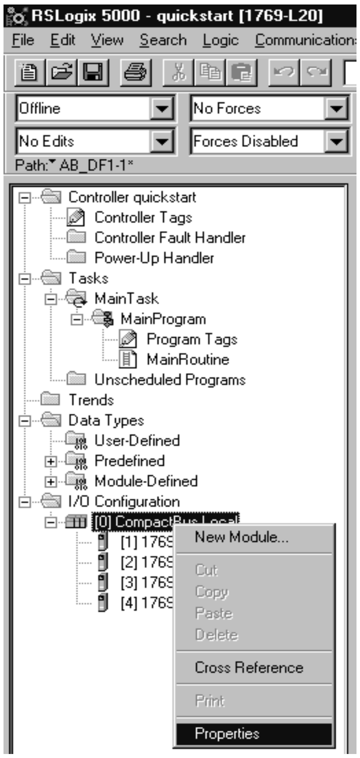
(1) Tag scope allows tag name uniqueness to be limited to a single scope (controller or program). Thus, the same tag name can be used in many programs without ambiguity.

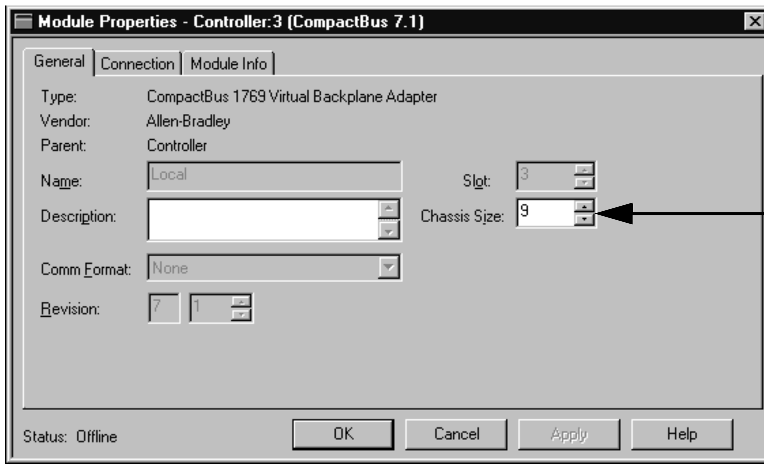
Configuring the CompactBus

3 **Configure the CompactBus**

1. In the Controller Organizer, select the CompactBus. Right-click and select Properties.

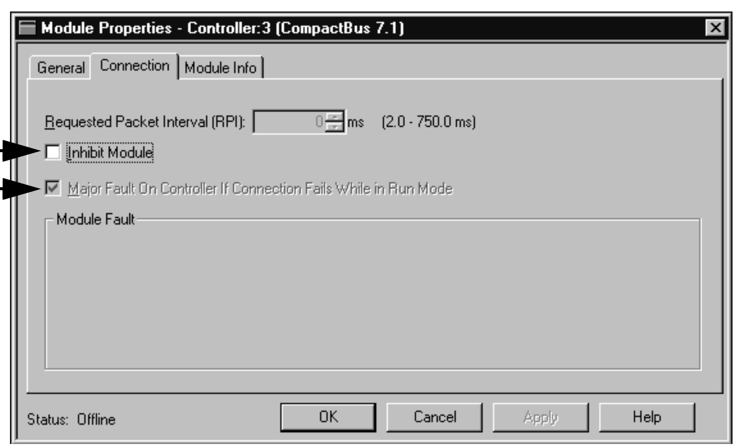
2. Specify the configuration options.





Specify the size of the chassis. Enter the number of modules you plan to install. Include the CompactLogix5320 controller in this total.

3. **View the Connection tab.**



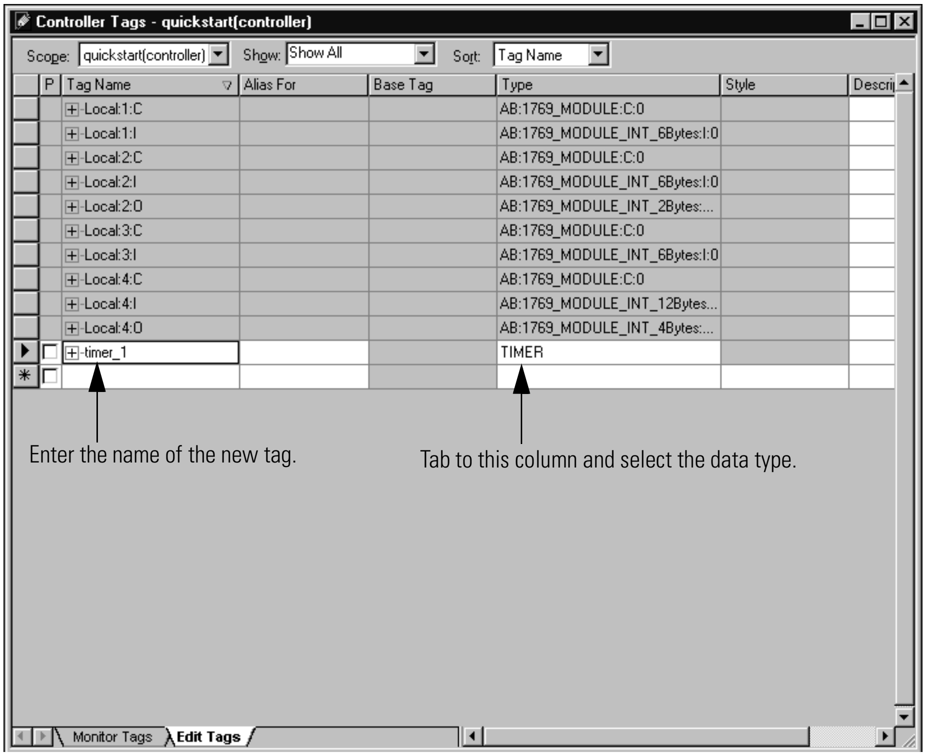
You can inhibit and uninhibit the CompactBus. By inhibiting and uninhibiting the CompactBus, you can write new configuration data to the entire system at once.

The controller's response to a CompactBus connection failure is fixed to always fault the controller. It is not configurable.

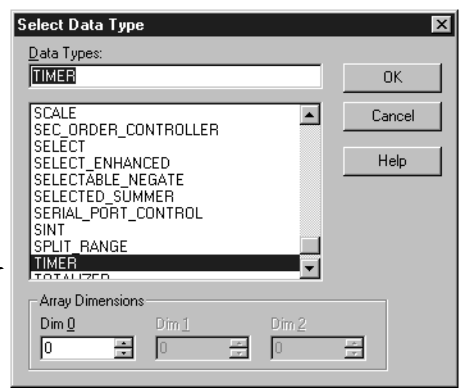
Creating Other Tags

4 **Create Tags**

1. **Click on *Edit tags* to create a tag.**



2. **Select the data type.**

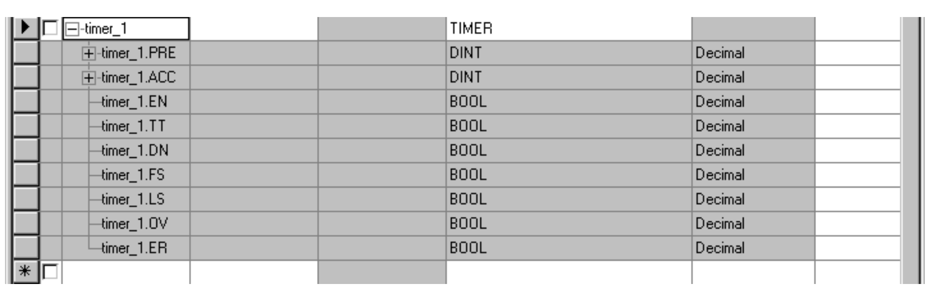


Select TIMER.

Click OK.

The software displays the tag.

Click + to display the members of the TIMER structure.



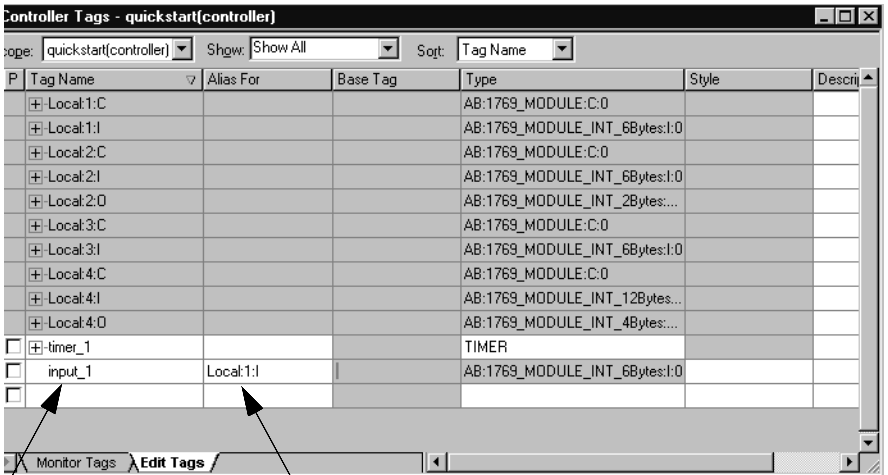
You might have to resize the column to see the tag extensions.

continued

4 **Create Tags**

Documenting I/O with Alias Tags

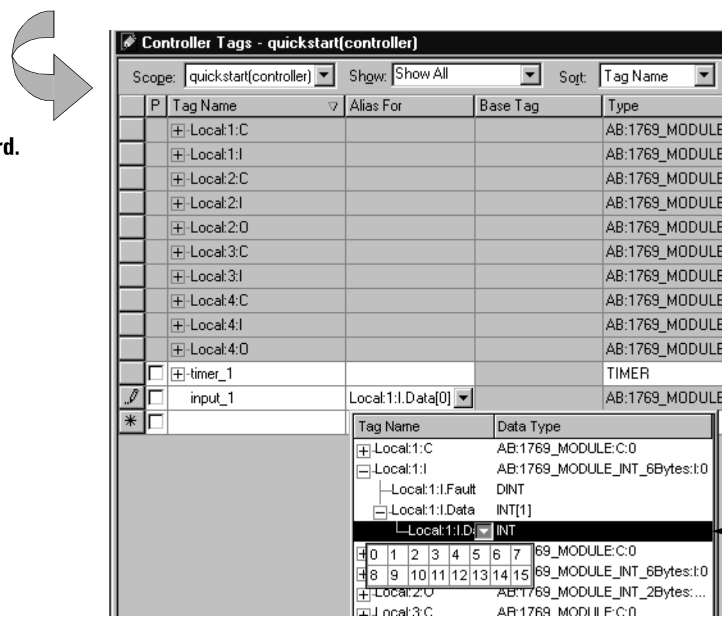
1. **Create an alias tag *input_1* for Local:1:I.Data.1.**



Enter the name of the tag.

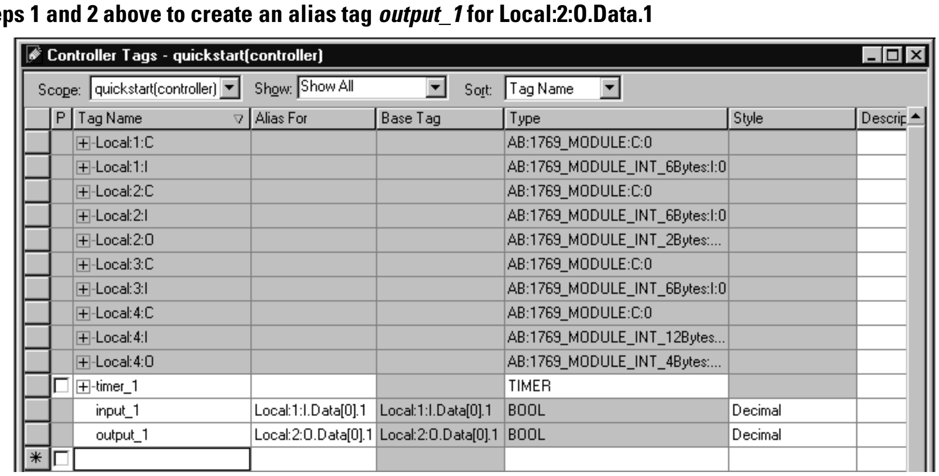
Tab here or click in the box. Click here to select tag to reference.

2. **Select an input data word.**



Click here to display a grid of bits and select the input bit.

3. **Repeat steps 1 and 2 above to create an alias tag *output_1* for Local:2:O.Data.1**

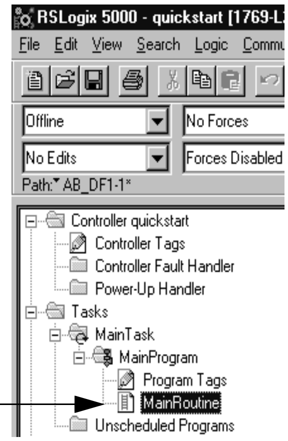


Entering Logic

5 **Enter Logic**

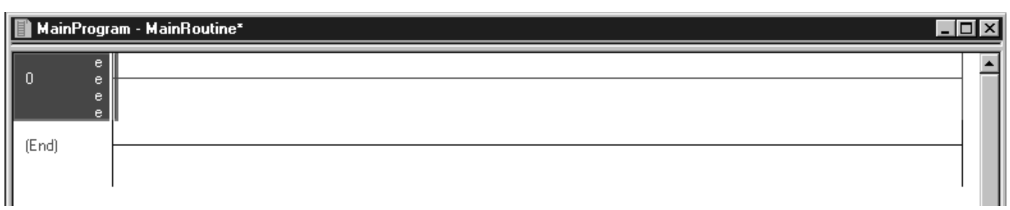
1. Use default task, program, and routine.

When you created the project, the software automatically created a MainTask, MainProgram, and MainRoutine. Use these defaults for this example.

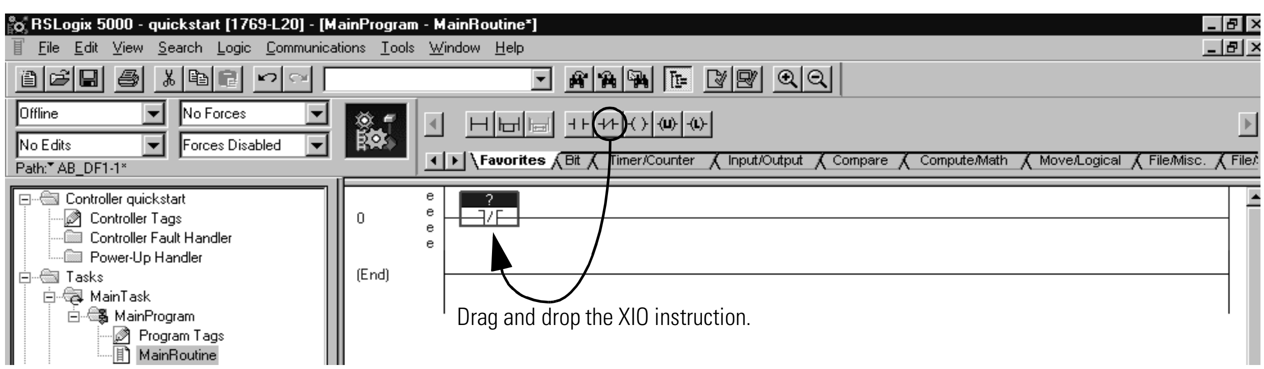


Double-click MainRoutine.

The software displays an empty routine.



2. Enter an XIO instruction.



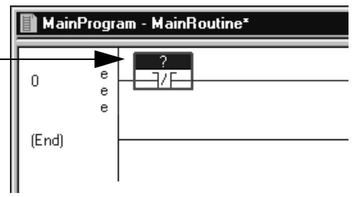
Drag and drop the XIO instruction.

Entering Logic (*continued*)

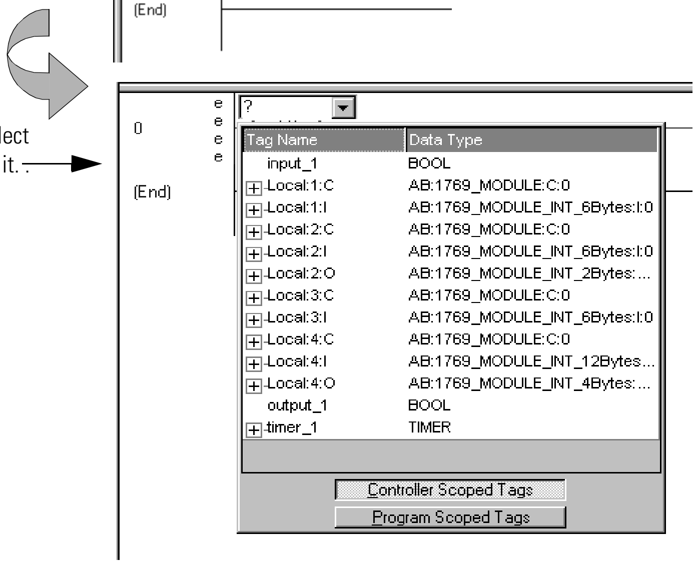
5 **Enter Logic**

3. **Assign a tag to the XIO instruction.**

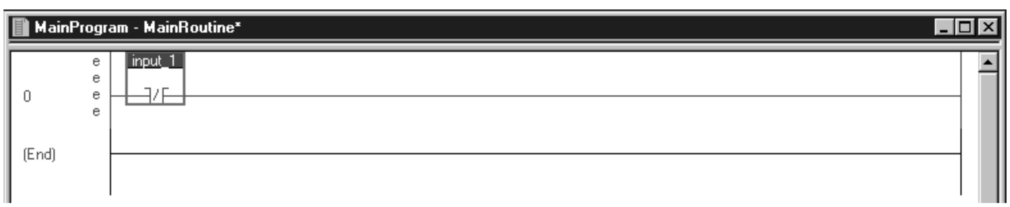
Double-click the tag area of the instruction.



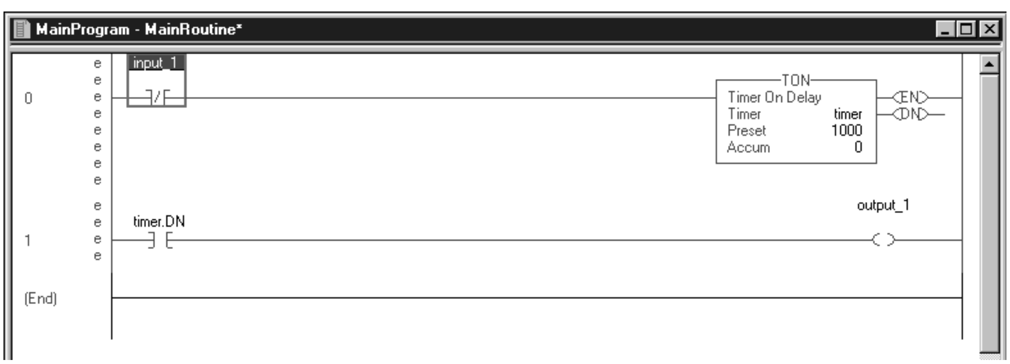
Use the drop-down menu to select *input_1* by double-clicking on it.



The software displays an incomplete rung.



4. **Enter this logic.**

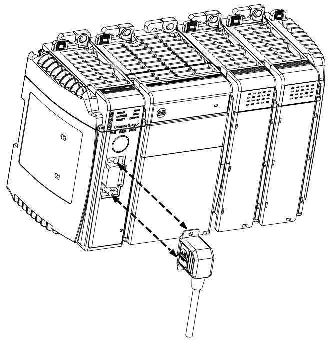


5. **To save the project, from the File menu, select Save.**

Downloading a Project

6 **Download a Project**

1. **Make a serial connection from the workstation to the controller.**



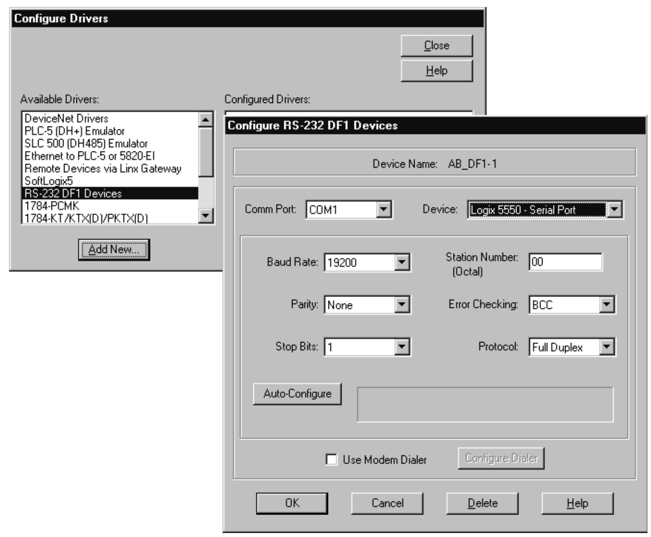
2. **Configure an RSLinx communication driver:**

A. In RSLinx software, select Communication → Configure Drivers.

B. From the Available Driver Types list, select "RS-232 DF1 Devices" and click Add New.

C. Choose a name for the driver and click OK.

D. Select the "Logix5550 serial port" from the Device pull-down menu, and specify the COM port. Click Auto-Configure to have the software determine the remaining serial settings.



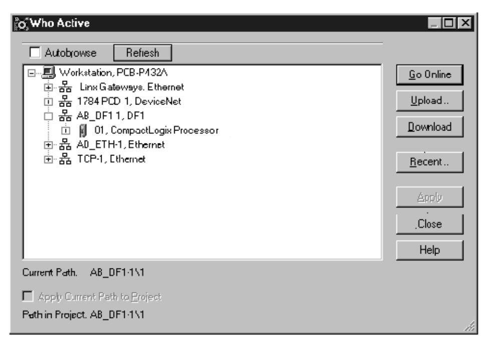
3. **Download the project from the Communications menu.**

A. Be sure the controller is in the Program mode.

B. In RSLogix5000 software, select Communication → Who Active.

C. Expand the DF1 network and select your controller.

D. Click Download. Confirm the download when prompted.



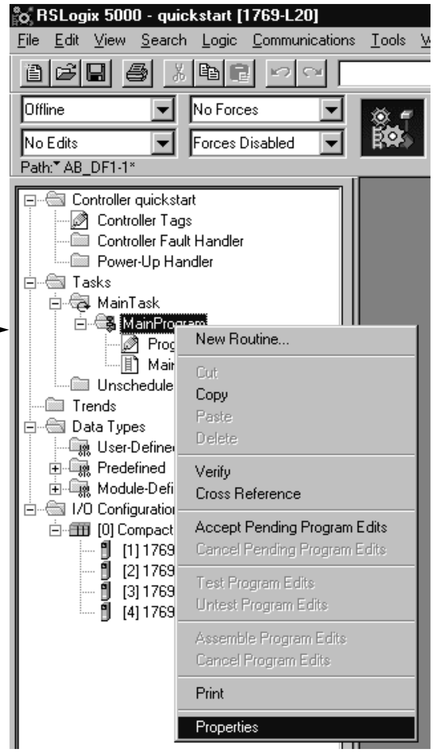
4. **Place the controller in Remote Run mode.**

Viewing Program Scan Time

1. View properties for the MainProgram.

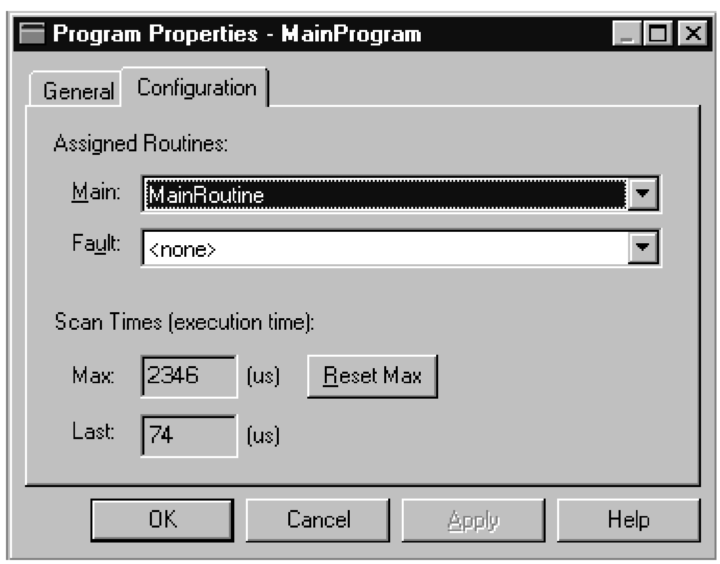
A. Place the cursor over the MainProgram folder.

B. Click the right mouse button and select Properties.



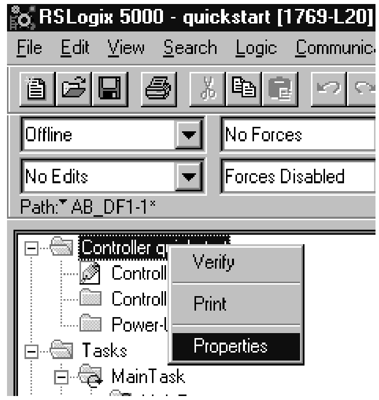
2. Select the Configuration tab.

The Configuration tab displays the maximum and last scan times for the program.



Viewing controller Memory Usage

1. **View properties for Controller quickstart.**

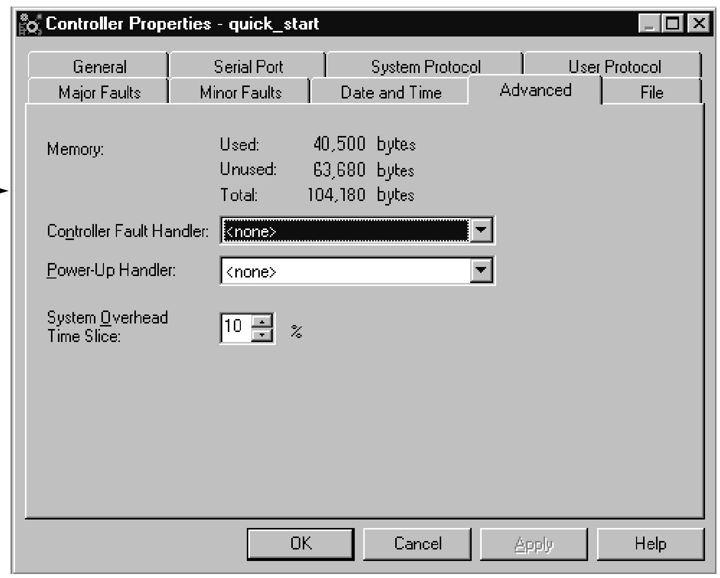


   A. Place the cursor over the Controller quickstart folder.

   B. Click the right mouse button and select Properties.

2. **Select the Advanced tab.**

In addition to other information, the Advanced tab displays controller memory usage.



IMPORTANT	The amount of memory that the software displays includes both the user-available memory and the memory reserved for overhead. Certain operations dynamically allocate and re-allocate user-available memory. See the specifications for your controller and Appendix E, *Dynamic Memory Allocation in CompactLogix Controllers*, to estimate how much memory you have available for programming.

What To Do Next

Once your controller is installed and operating, you can use RSLogix 5000 programming software to develop and test your control application.

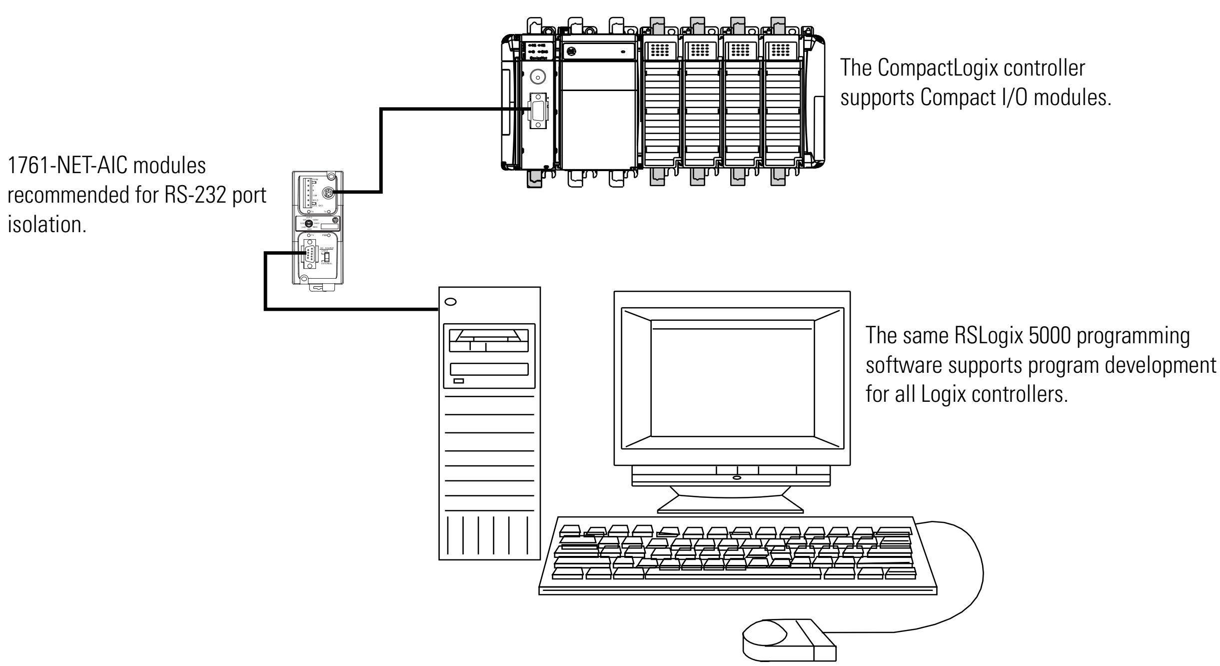
Use the remaining chapters in this manual as reference material for how the CompactLogix controller operates in the Logix environment.

What Is CompactLogix?

Using This Chapter

The CompactLogix controller, part of the Logix family of controllers, provides a small, cost-effective system built on these components:

- CompactLogix controller that supports the Logix instruction set.
- RSLogix 5000 programming software that supports every Logix controller.
- Compact I/O modules that provide a compact, DIN-rail or panel-mounted I/O system.
- Serial port that supports multiple communication protocols.
- Communication interface modules provide peer-to-peer communication over DH-485, DeviceNet, or Ethernet. DeviceNet and Ethernet support program upload/download.

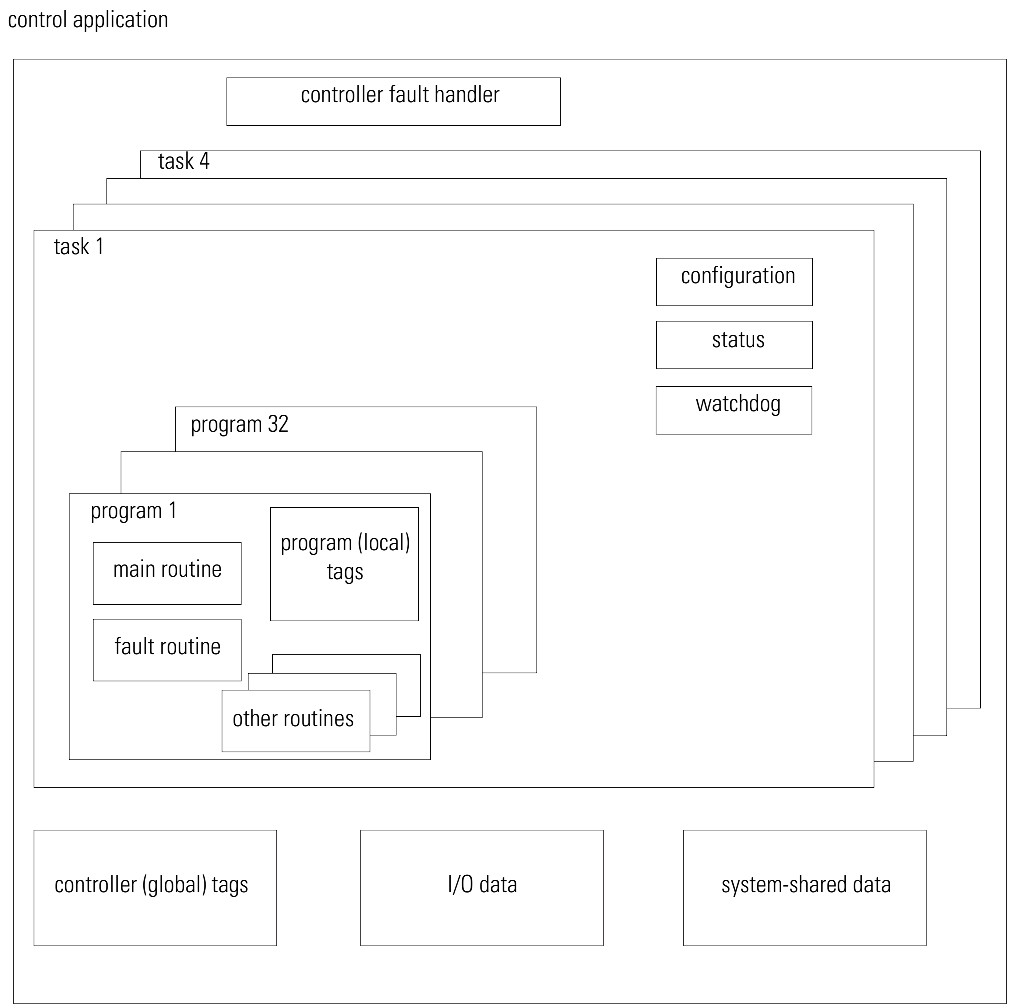


For information about:	See page
developing programs	2-2
direct connections for I/O modules	2-6
selecting a system overhead percentage	2-7

Developing Programs

The controller operating system is a preemptive multitasking system that is IEC 1131-3 compliant. This environment provides:

- tasks to configure controller execution
- programs to group data and logic
- routines to encapsulate executable code written in a single programming language



Defining Tasks

A task provides scheduling and priority information for a set of one or more programs. You can configure tasks as either continuous or periodic. The CompactLogix controller supports as many as 4 tasks, only one of which can be continuous.

A task can have as many as 32 separate programs, each with its own executable routines and program-scoped tags. Once a task is triggered (activated), all the programs assigned to the task execute in the order in which they are grouped. Programs can only appear once in the Controller Organizer and cannot be shared by multiple tasks.

Specifying Task Priorities

Each task in the controller has a priority level. The operating system uses the priority level to determine which task to execute when multiple tasks are triggered. You can configure periodic tasks to execute from the lowest priority of 15 up to the highest priority of 1. A higher priority task will interrupt any lower priority task. The continuous task has the lowest priority and is always interrupted by a periodic task.

The CompactLogix controller uses a dedicated periodic task at priority 7 to process I/O data. This periodic task executes every 2 ms. Its total execution time is as long as it takes to scan the configured I/O modules. A typical I/O scan of eight discrete I/O modules for a CompactLogix5320 controller is approximately 400 μs.

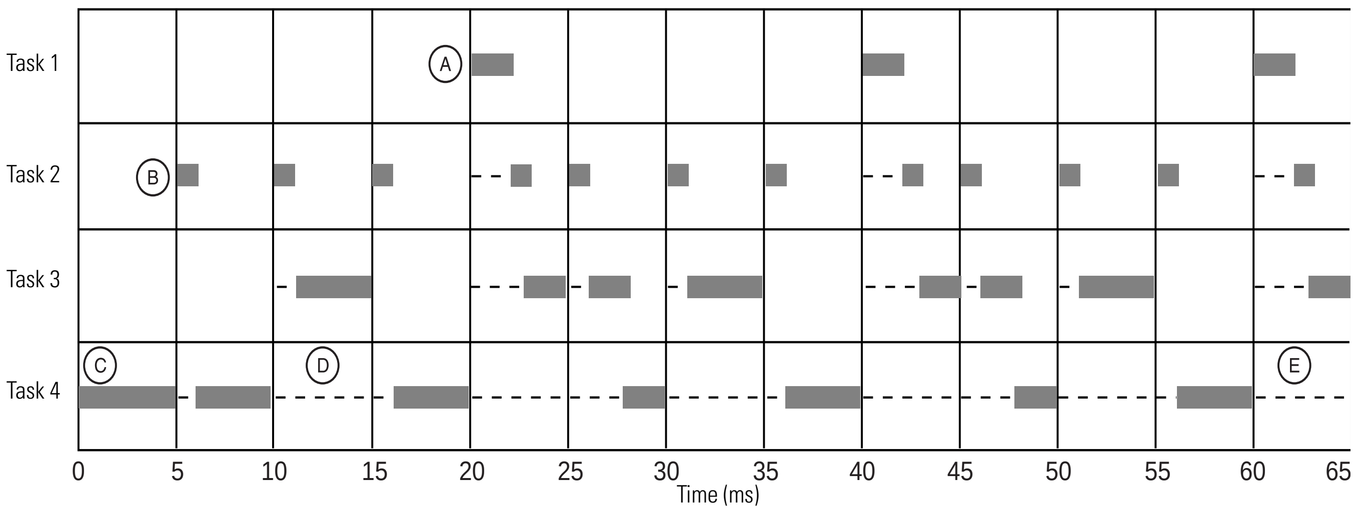
How you configure your tasks affects how the controller receives I/O data. Tasks at priorities 1 to 6 take precedence over the dedicated I/O task. Tasks in this priority range can impact I/O processing time. A task of priority 1 to 6 that requires 1/2 ms to execute and is scheduled to run every millisecond, consumes 1 ms of CPU time. This leaves the dedicated I/O task 1 ms to complete its job of scanning the configured I/O. However, if you were to schedule two high priority tasks (1 to 6) to run every millisecond, and they both required 1/2 ms or more to execute, no CPU time would be left for the dedicated I/O task . Furthermore, if you have so much configured I/O that the execution time of the dedicated I/O task approaches 2 ms (or the combination of the high priority tasks and the dedicated I/O task approaches 2 ms) no CPU time is left for low priority tasks (8 to 15).

NOTE

For example, if your program needs to react to inputs and control outputs at a deterministic rate, configure a periodic task with a priority higher than 7 (1 through 6). This keeps the dedicated I/O task from affecting the periodic rate of your program. However, if your program contains a lot of math and data manipulation, place this logic in a task with priority lower than 7 (8 through 15), such as the continuous task, so that the dedicated I/O task is not adversely affected by your program.

The following example shows the task execution order for an application with periodic tasks and a continuous task.

Task:	Priority Level:	Task Type:	Example Execution Time:	Worst Case Completion Time:
1	5	20 ms periodic task	2 ms	2 ms
2	7	dedicated I/O task 2 ms fixed RPI	1 ms	3 ms
3	10	10 ms periodic task	4 ms	8 ms
4	none (lowest)	continuous task	25 ms	60 ms



Notes:

A. The highest priority task interrupts all lower priority tasks.

B. The dedicated I/O task can be interrupted by tasks with priority levels 1 to 6. The dedicated I/O task interrupts tasks with priority levels 8 to 15. This task runs at the fixed RPI rate scheduled for the CompactLogix system (2ms in this example).

C. The continuous task runs at the lowest priority and is interrupted by all other tasks.

D. A lower priority task can be interrupted multiple times by a higher priority task.

E. When the continuous task completes a full scan it restarts immediately, unless a higher priority task is running.

Defining Programs

Each program contains program tags, a main executable routine, other routines, and an optional fault routine. Each task can schedule as many as 32 programs.

The scheduled programs within a task execute to completion from first to last. Programs that aren't attached to any task show up as unscheduled programs. You must specify (schedule) a program within a task before the controller can scan the program.

Defining Routines

A routine is a set of logic instructions in a single programming language, such as ladder logic. Routines provide the executable code for the project in a controller. A routine is similar to a program file or subroutine in a PLC or SLC controller.

Each program has a main routine. This is the first routine to execute when the controller triggers the associated task and calls the associated program. Use logic, such as the Jump to Subroutine (JSR) instruction, to call other routines.

You can also specify an optional program fault routine. The controller executes this routine if it encounters an instruction-execution fault within any of the routines in the associated program.

Direct Connections for I/O Modules

The CompactLogix system uses direct connections to transmit I/O data. Each local I/O module utilizes a direct connection to the CompactLogix controller. A direct connection is a real-time, data transfer link between the controller and an I/O module. The controller maintains and monitors the connection between the controller and the I/O module. Any break in the connection, such as a module fault, causes the controller to set fault status bits in the input data area associated with the module.

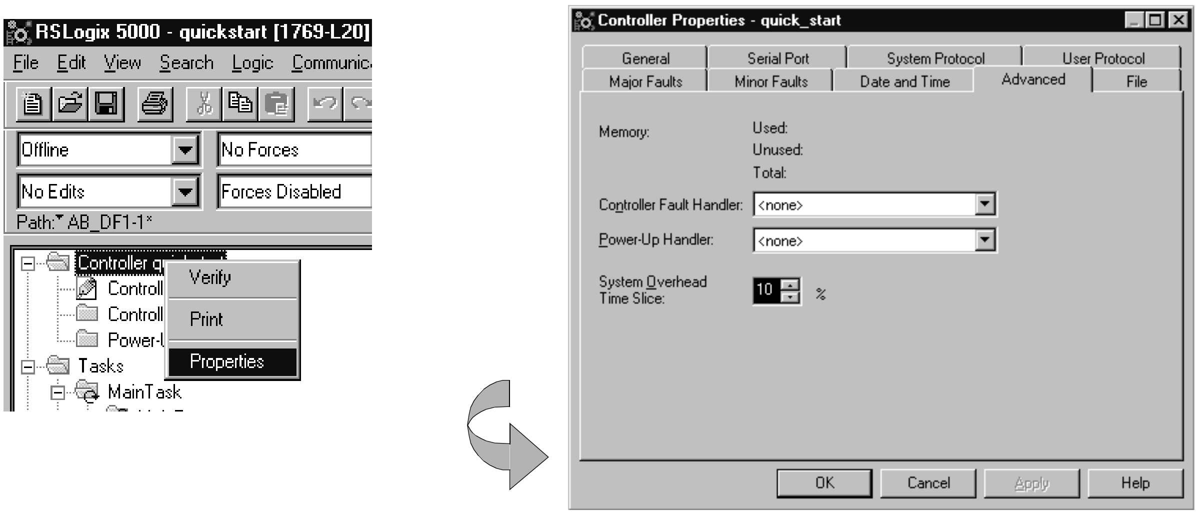
ATTENTION

CompactLogix does not support Removal and Insertion Under Power (RIUP). While the CompactLogix system is under power, any break in the connection between the power supply and the processor (i.e. removing the power supply, processor, or an I/O module) will clear processor memory (including the user program).

Selecting a System Overhead Percentage

The Controller Properties lets you specify a percentage for system overhead. This percentage specifies the percentage of controller time (excluding the time for periodic tasks) that is devoted to communication and background functions

1. View properties for the controller and select the Advanced tab.



System overhead functions include:

- communicating with programming and HMI devices (such as RSLogix 5000 software)
- responding to messages
- sending messages, including block-transfers

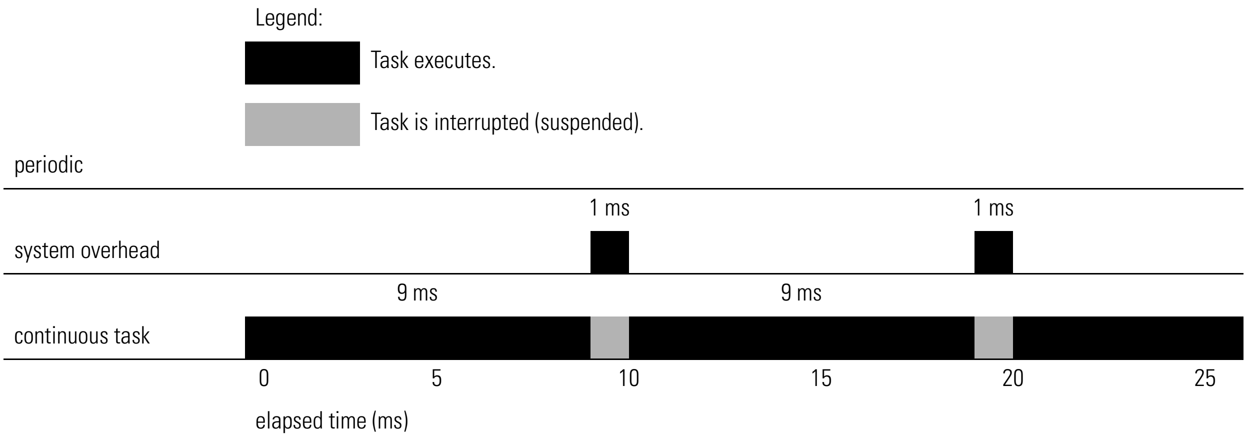
The controller performs system overhead functions for up to 1 ms at a time. If the controller completes the overhead functions in less than 1 ms, it resumes the continuous task.

If communications are not completing fast enough, increase the system overhead percentage. As you increase the system overhead percentage, the overall program scan also increases.

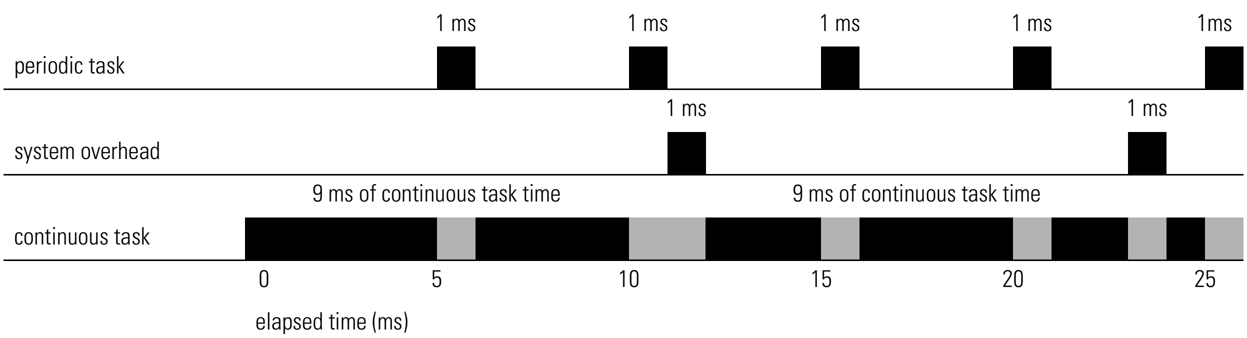
The following table shows the ratio between the continuous task and the system overhead functions:

At this time slice:	The continuous tasks runs for:	And then overhead occurs for up to:
10%	9 ms	1 ms
20%	4 ms	1 ms
33%	2 ms	1 ms
50%	1 ms	1 ms

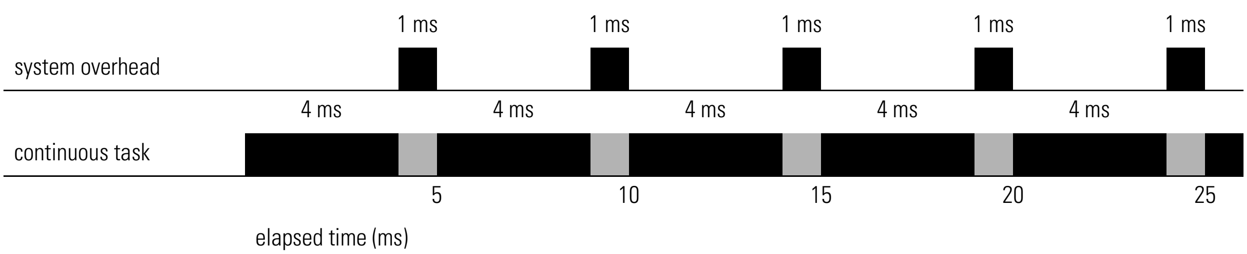
At the default time slice of 10%, system overhead interrupts the continuous task every 9 ms (of continuous task time), as illustrated below.



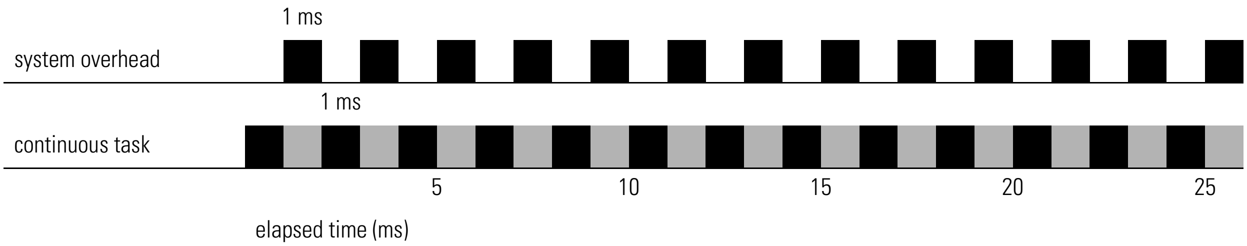
The interruption of a periodic task increases the elapsed time (clock time) between the execution of system overhead, as shown below.



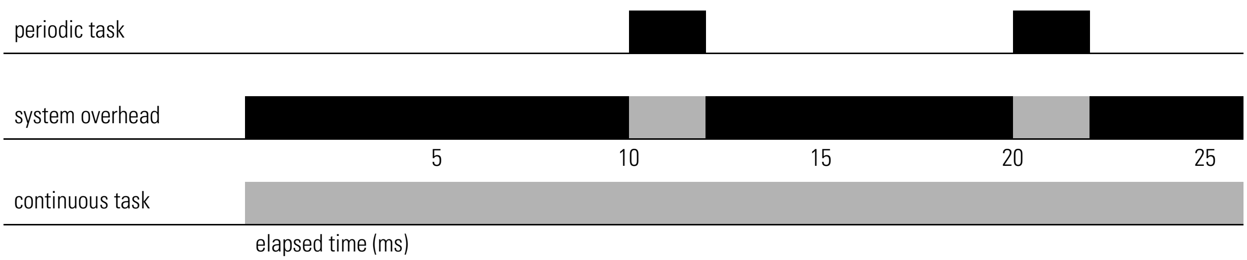
If you increase the time slice to 20%, the system overhead interrupts the continuous task every 4 ms (of continuous task time).



If you increase the time slice to 50%, the system overhead interrupts the continuous task every 1 ms (of continuous task time).



If the controller only contains a periodic task(s), the system overhead timeslice value has no effect. System overhead runs whenever a periodic task is not running.



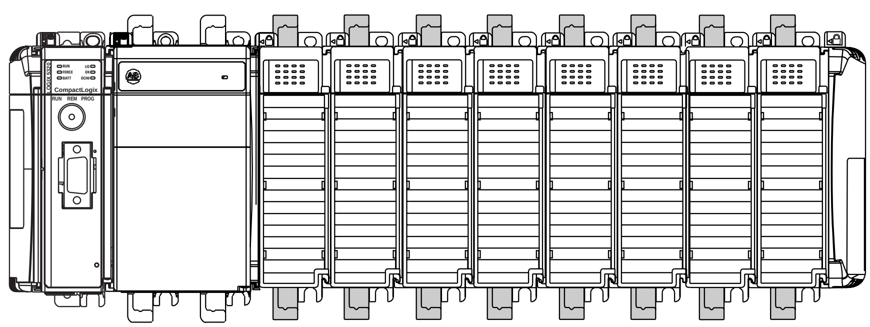
Placing, Configuring, and Monitoring Local I/O

Using This Chapter

Placing Local I/O Modules

I/O Configuration Rules

The CompactLogix controller supports as many as 8 local I/O modules on the CompactBus.

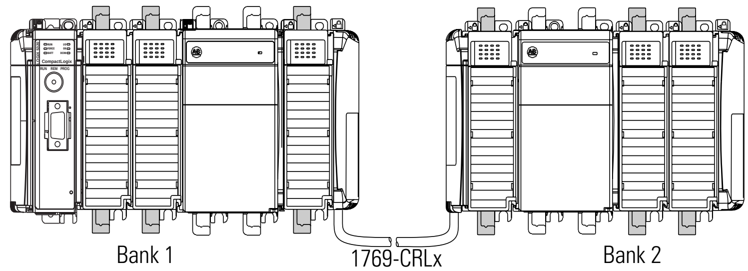


You can also use the 1769-CRR1/-CRR3 or 1769-CRL1/-CRL3 cable to split the system into a maximum of two banks of I/O modules. You can split a bank right after the power supply or after any I/O module. Each bank must contain one power supply. An end cap/terminator must be used on the last I/O bank.

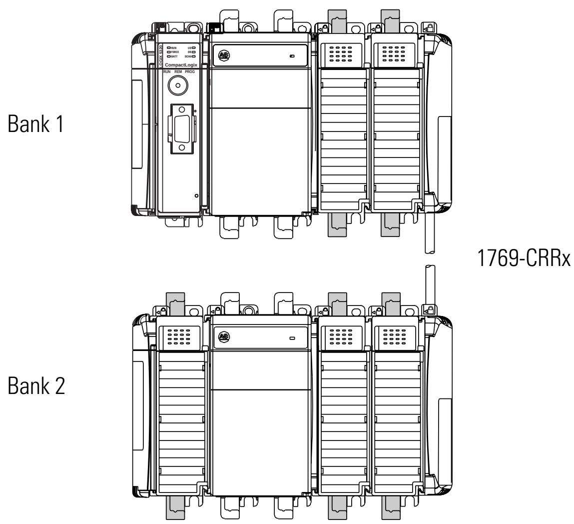
The first bank includes the CompactLogix controller in the far left position. The controller must be located within 4 positions of the bank's power supply. Only one controller may be used in a CompactLogix system.

Each I/O module also has a power supply distance rating (the number of modules from the power supply). Each module must be located within its distance rating.

Horizontal Orientation



Vertical Orientation



ATTENTION

CompactLogix does not support Removal and Insertion Under Power (RIUP). While the CompactLogix system is under power, any break in the connection between the power supply and the processor (i.e. removing the power supply, processor, or an I/O module) will clear processor memory (including the user program).

IMPORTANT

While under power, the removal of an end cap or a module (without breaking the connection between the processor and power supply) will fault the controller.

If the controller was in Program mode, cycle power in order to go to run successfully.

If the controller was in Run mode, cycle power. When the I/O LED is on (green, steady) and the OK LED is flashing red, turn the keyswitch from Program to Run and back to Program. When the OK LED turns on (green, steady), turn the keyswitch to Run.

System Power Budget Calculation and Considerations

To validate your system, the total 5V dc current and 24V dc current consumed must be considered. The I/O modules must be distributed such that the current consumed from the left or right side of the power supply never exceeds 2.0A at 5V dc and 1.0A at 24V dc.

Use the following worksheet as a general guide to account for the amount of 5V dc and 24V dc current consumed by each band of I/O in your system. Be sure to follow the I/O configuration rules on page 3-1 when planning your system.

Catalog Number	Number of Modules	Module Current Requirements		Calculated Current = (Number of Modules) x (Module Current Requirements)	
		at 5V dc (in mA)	at 24V dc (in mA)	at 5V dc (in mA)	at 24V dc (in mA)
1769-L20		600	0		
1769-IA16		115	0		
1769-IA8I		90	0		
1769-IM12		100	0		
1769-IQ16		115	0		
1769-IQ6XOW4		105	50		
1769-OA8		145	0		
1769-OB16		200	0		
1769-OB16P		180	0		
1769-OV16		200	0		
1769-OW8		125	100		
1769-OW8I		125	100		
1769-IF4 (A)		120	150		
1769-IF4 (B)		120	60		
1769-OF2 (A)		120	200		
1769-OF2 (B)		120	120		
1769-ECR(1)		5	0		
1769-ECL(1)		5	0		
Total Modules:		**Total Current Required(2):**			

(1) One 1769-ECR or 1769-ECL end cap/terminator is required in the system. The end cap/terminator used is dependent on your configuration.
(2) This number must not exceed the Power Supply Current Capacity listed below.

Power Supply Current Capacity

Specification	1769-PA2	1769-PB2	1769-PA4	1769-PB4
Output Bus Current Capacity (0°C to +55°C)	2A at 5V dc and 0.8A at 24V dc		4A at 5V dc and 2A at 24V dc	
24V dc User Power Capacity (0°C to +55°C)	250 mA (maximum)	not applicable		

Determining When the Controller Updates I/O

The controller continually scans the control logic. One scan is the time it takes the controller to execute the logic once. Input data transfers to the controller and output data transfers to output modules are asynchronous to the logic scan.

NOTE

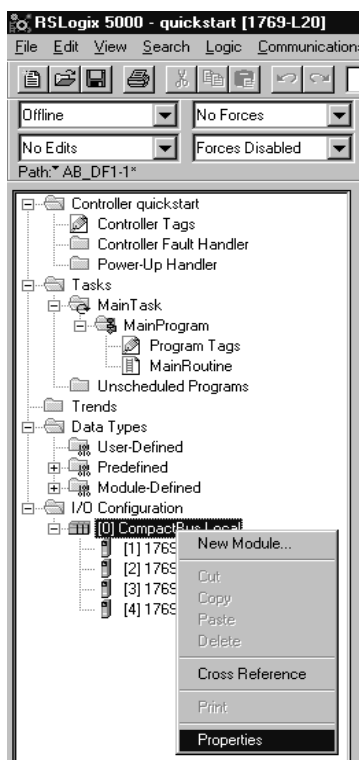
If you need to synchronize I/O to the logic scan, you can use the Synchronous Copy instruction (CPS) to buffer the I/O data.

Refer to the *Logix5000 Controllers Common Procedures Programming Manual*, publication number 1756-PM001B-EN-P for examples of I/O buffering or to the *Logix5000 Controllers General Instruction Set Reference Manual,* publication number 1756-RM003C-EN-P for information on the CPS instruction.

Configuring the CompactBus

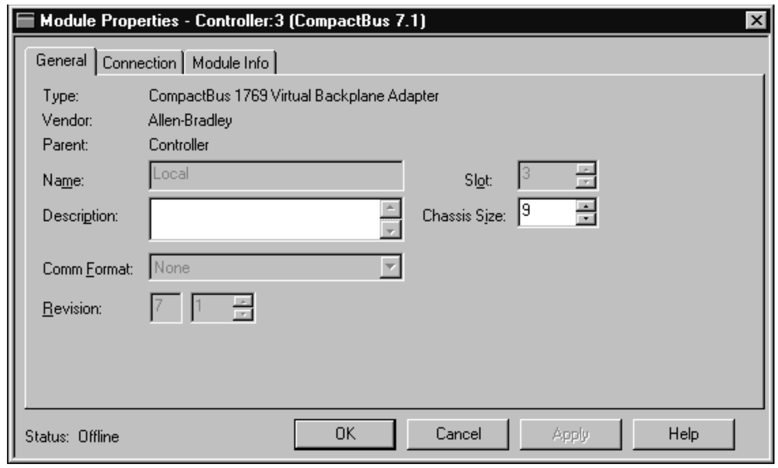
When you create a CompactLogix project, the programming software automatically creates the local CompactBus. You must configure the CompactBus.

To access the CompactBus properties screens, select the local CompactBus from the Controller Organizer. Right-click and select Properties.



Specifying General Properties

On the General tab, specify the size of the Chassis. Enter the number of modules you plan to install. Include the CompactLogix5320 controller in this total, along with a maximum of 8 I/O modules.

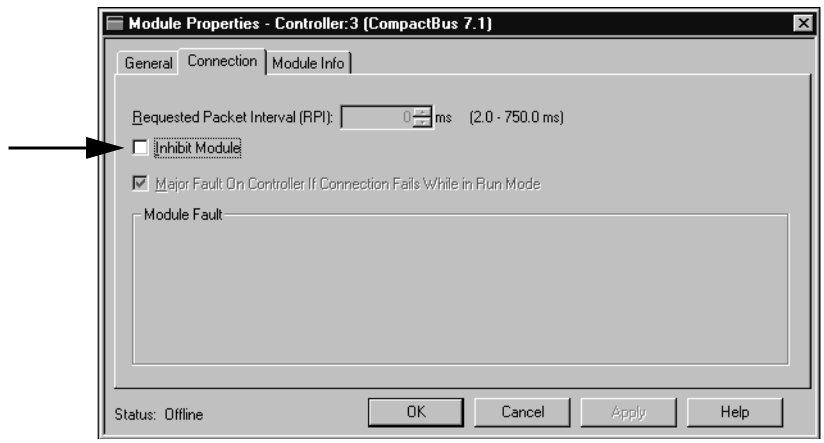


NOTE The Comm Format for the CompactBus is automatically set to None and cannot be changed because the controller uses direct connections to each I/O module.

Inhibiting the CompactBus

Using the Connection tab, you can choose to inhibit or uninhibit the CompactBus.

NOTE By inhibiting and then uninhibiting the CompactBus, you can write new configuration data to all modules in the system at once.



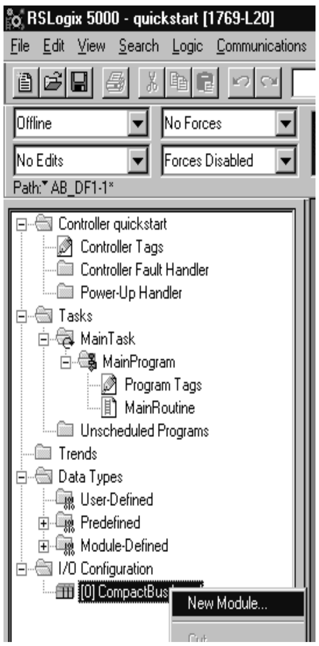
IMPORTANT The controller's response to a CompactBus connection failure is fixed to always fault the controller. It is not configurable.

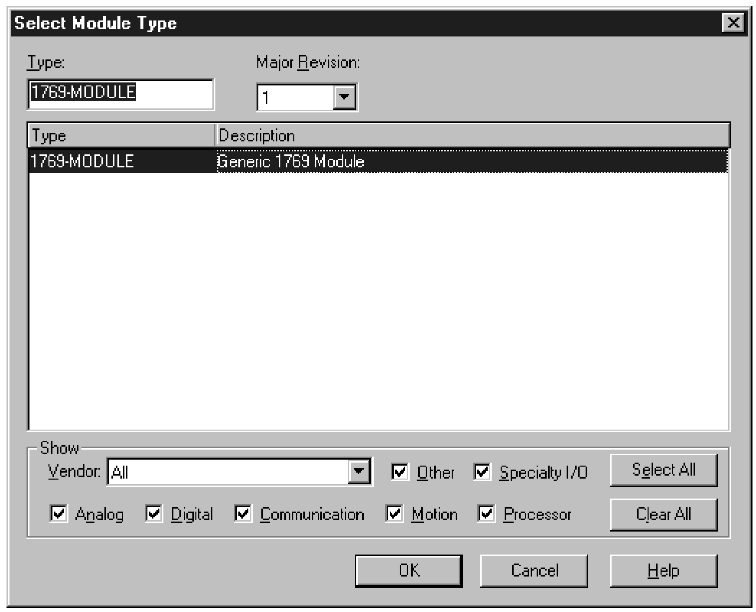
Configuring Local I/O Modules

Use your programming software to configure the I/O modules for the controller. All the I/O modules on the CompactBus operate at a fixed RPI of 2 ms.

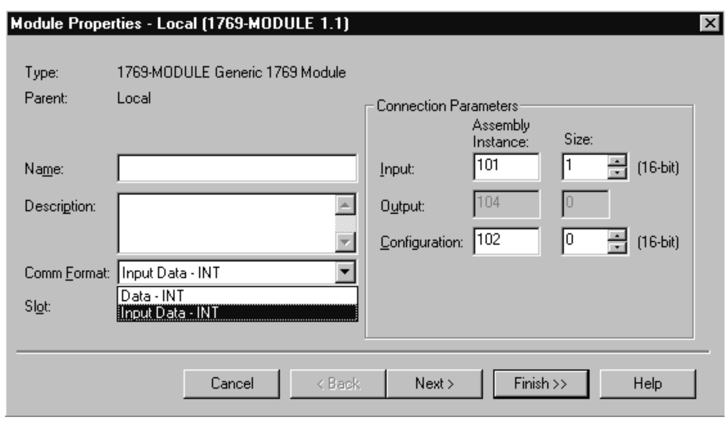
To configure an I/O module, right-click on the local CompactBus in the Controller Organizer and select New Module.

Select the Generic 1769 Module. For additional information and an example, see Appendix G, *Configuring I/O with the 1769 Generic Profile*.





Enter the appropriate connection parameters as well as the comm format and slot number and name for the module. See the explanation of comm formats on page 3-8.



NOTE The selection you make for the Comm Format determines the connections required for the I/O module. Once you complete adding a module, you cannot change this selection.

Communication Formats

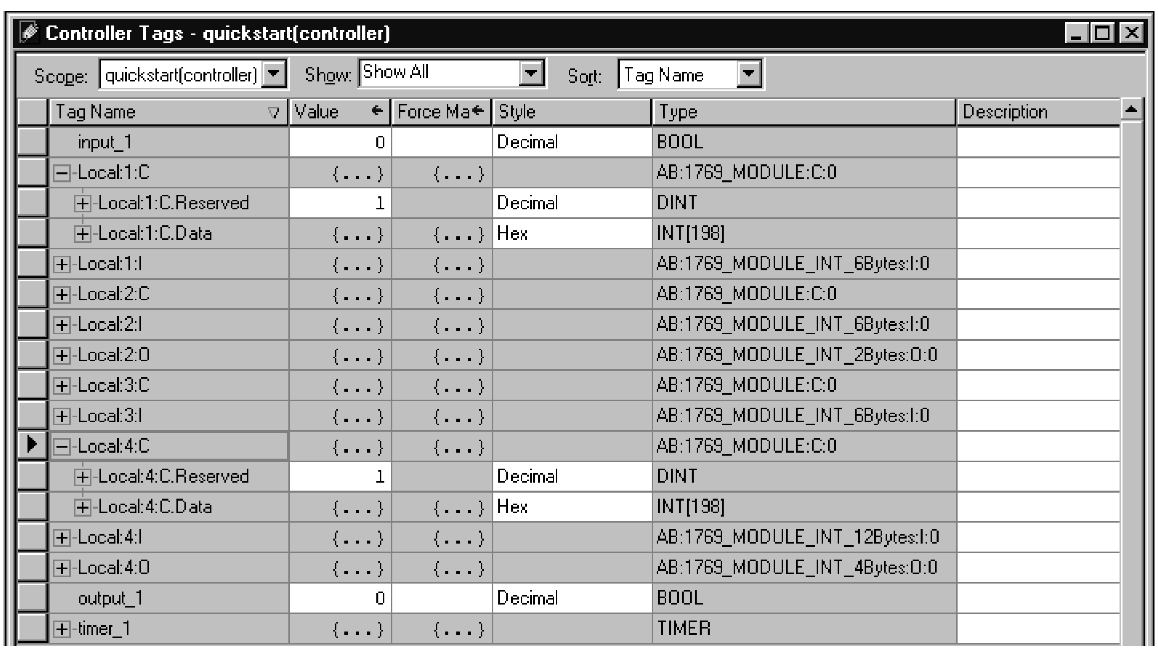
The communication format determines the data structure the I/O module uses. Each format supports a different data structure. Presently, the CompactLogix system supports two data formats:

- Input Data – INT (for 1769 input modules)
- Data – INT (for 1769 output modules)

You select the communications format when you create the I/O module. The default communication format for a generic profile module is Input Data-INT.

NOTE The CompactLogix controller must own its local I/O modules. No other Logix-based controller can own the local CompactLogix I/O.

The communication format determines the tag structure that is created for the module. Assume that a 1769-IB16 Input module is in slot 1. The software creates the appropriate tags using the slot number to differentiate the tags for this example module from any other module.


Controller Tags - quickstart(controller)

Scope: quickstart(controller) Show: Show All Sort: Tag Name

Tag Name	Value	Force Ma	Style	Type	Description
input_1	0		Decimal	BOOL	
-Local:1:C	{...}	{...}		AB:1769_MODULE:C:0	
+Local:1:C.Reserved	1		Decimal	DINT	
+Local:1:C.Data	{...}	{...}	Hex	INT[198]	
+Local:1:I	{...}	{...}		AB:1769_MODULE_INT_6Bytes:I:0	
+Local:2:C	{...}	{...}		AB:1769_MODULE:C:0	
+Local:2:I	{...}	{...}		AB:1769_MODULE_INT_6Bytes:I:0	
+Local:2:O	{...}	{...}		AB:1769_MODULE_INT_2Bytes:O:0	
+Local:3:C	{...}	{...}		AB:1769_MODULE:C:0	
+Local:3:I	{...}	{...}		AB:1769_MODULE_INT_6Bytes:I:0	
-Local:4:C	{...}	{...}		AB:1769_MODULE:C:0	
+Local:4:C.Reserved	1		Decimal	DINT	
+Local:4:C.Data	{...}	{...}	Hex	INT[198]	
+Local:4:I	{...}	{...}		AB:1769_MODULE_INT_12Bytes:I:0	
+Local:4:O	{...}	{...}		AB:1769_MODULE_INT_4Bytes:O:0	
output_1	0		Decimal	BOOL	
+timer_1	{...}	{...}		TIMER	

See Appendix G, *Configuring I/O with the 1769 Generic Profile*, for details on the data structures created by the Generic module.

Inhibiting I/O Module Operation

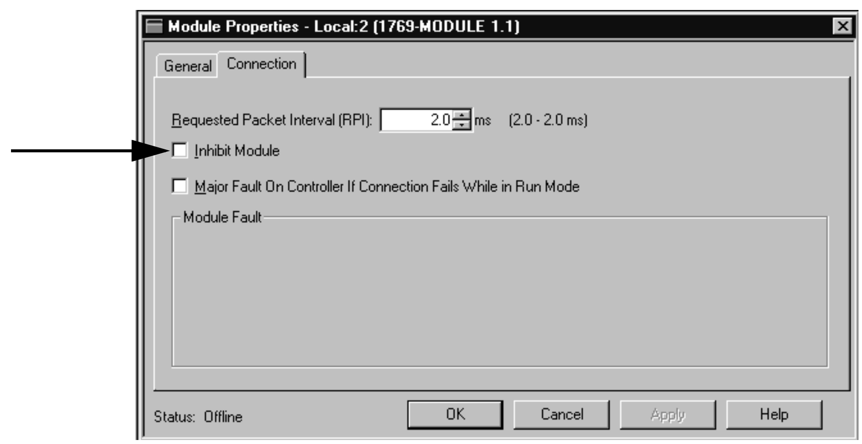
In some situations, such as when initially commissioning a system, it is useful to disable portions of a control system and enable them as you wire up the control system. The controller lets you inhibit individual modules or groups of modules, which prevents the controller from trying to communicate with the modules. Inhibiting a module shuts down the connection from the controller to that module.

When you create an I/O module, it defaults to being not inhibited. You can change an individual module's properties to inhibit a module.

ATTENTION

Inhibiting a module closes the connection to the module and prevents communication of I/O data.

On the Connection tab of the Module Properties dialog, you can select to inhibit that specific module.



NOTE

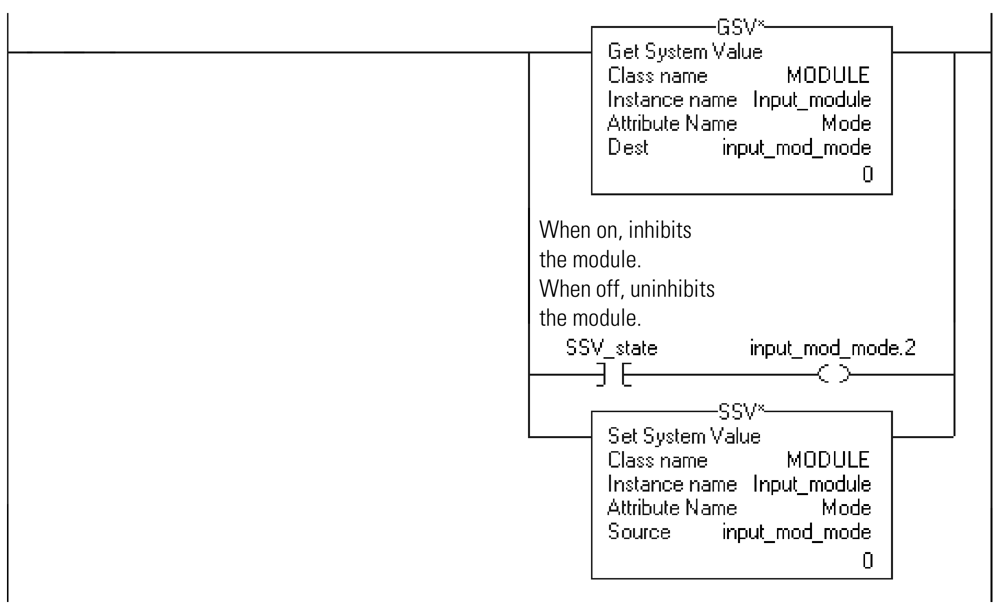
To easily inhibit all local I/O modules, you can inhibit the CompactBus, which in turns inhibits all the modules on that bus. See *Configuring the CompactBus* on page 3-4.

When you select to inhibit a module, the controller organizer displays a yellow attention symbol ⚠ over the module.

If you are:	Inhibit a module to:
offline	put a place holder for a module you are configuring. The inhibit status is stored in the project. When you download the project, the module is still inhibited.
online	stop communication to a module. If you inhibit a module while you are connected to the module, the connection to the module is closed. The modules' outputs go to the last configured state for your program mode selections (Hold Last State or user-defined safe state). If you inhibit a module but a connection to the module was not established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is inhibited and not faulted. If you uninhibit a module (clear the check box), and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured with the configuration you created for that module. If you uninhibit the module and a fault condition occurs, a connection is not made to the module. The module status information changes to indicate the fault condition.

NOTE To inhibit a module from logic, you must first read the Mode attribute for the module using a GSV instruction. Set bit 2 to the inhibit status (1 to inhibit or 0 to uninhibit). Use a SSV instruction to write the Mode attribute back to the module. For example:

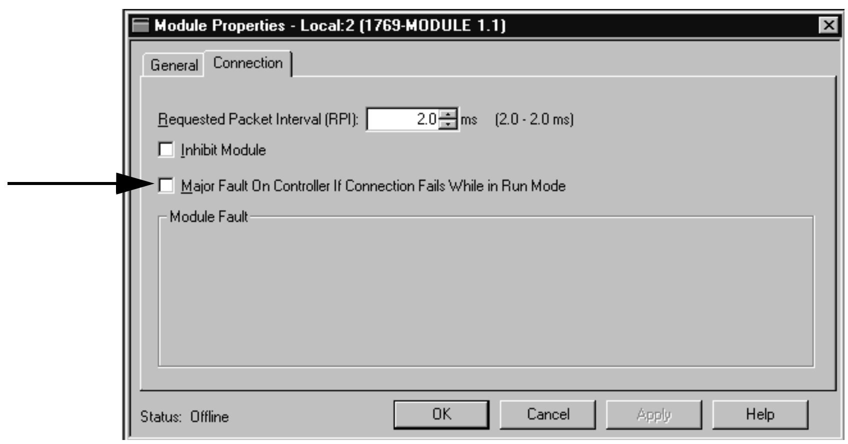
The GSV instruction gets the current status of the module named "input_module." The SSV instruction sets the state of "input_module" as either inhibited or uninhibited.



Configuring the Module's Response to a Connection Failure

Using the Connection tab in the ControlLogix and FlexLogix systems, you can also configure modules to generate a major fault in the controller if they lose their connection with the controller. This feature, however, is not available in a CompactLogix system because the controller's response to a CompactBus connection failure is fixed to always fault the controller. The CompactBus setting supersedes the individual module's setting. See *Inhibiting the CompactBus* on page 3-6.

IMPORTANT	Although you can select and de-select the "Major Fault On Controller If Connection Fails While in Run Mode" option, this will have no effect on controller operation. The controller's response to a connection failure of any I/O module is fixed to always fault the controller.



Accessing I/O Data

The programming software displays I/O data as structures of multiple tags that depend on the specific features of the I/O module. The names of the data structures are based on the location of the I/O module. The programming software automatically creates the necessary structures and tags when you configure the module. Each tag name follows this format:

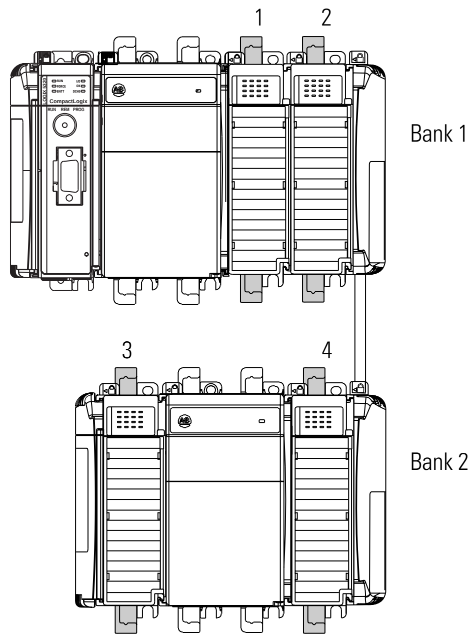
Location:SlotNumber:Type.MemberName.SubMemberName.Bit

where:

This address variable:	Is:
Location	Identifies network location LOCAL = local chassis
SlotNumber	Slot number of I/O module in its chassis
Type	Type of data I = input O = output C = configuration
MemberName	Specific data from the I/O module; depends on the type of data the module can store For example, Data and Fault are possible fields of data for an I/O module. Data is the common name for values the are sent to or received from I/O points.
SubMemberName	Specific data related to a MemberName.
Bit (optional)	Specific point on the I/O module; depends on the size of the I/O module (0-31 for a 32-point module)

The following examples show addresses for data in a CompactLogix system.

EXAMPLE I/O module on the local CompactBus utilizing two banks



Sample tag names for this example:

Location:	Example Tag Name:
input module in slot 1, LOCAL Bank 1	Local:1:C Local:1:I
output module in slot 2, LOCAL Bank 1	Local:2:C Local:2:I Local:2:O
analog input module in slot 3, LOCAL Bank 2	Local:3:C Local:3:I
analog output module in slot 4, LOCAL Bank 2	Local:4:C Local:4:I Local:4:O

Using Aliases to Simplify Tag Names

An alias lets you create a tag that represents another tag. This is useful for defining descriptive tag names for I/O values. For example:

Example:		**Description:**
I/O structure	Local:1:I:Data[0].0 Local:1:I:Fault.0	The aliases describe the specific I/O points.
alias	light_on = Local:1:I:Data[0].0 light_off = Local:1:I:Fault.0	

Monitoring I/O Modules

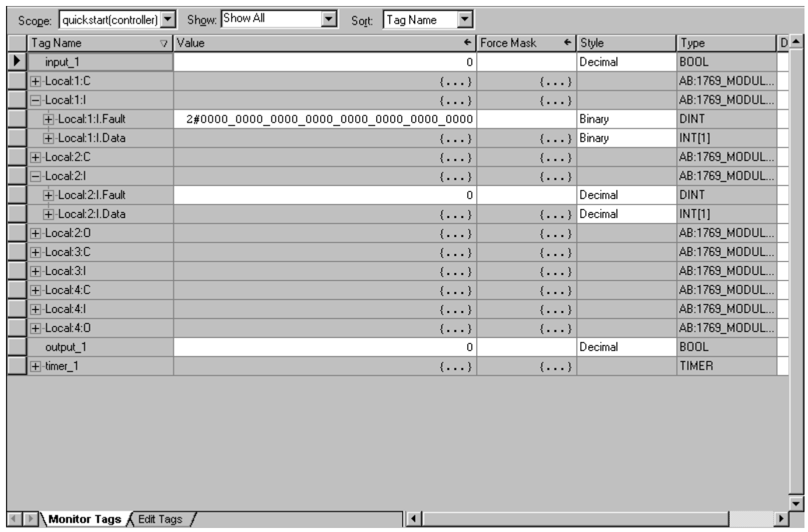
The CompactLogix controller offers different levels at which you can monitor I/O modules. You can:

- configure the module to fault if that module loses its connection to the controller (See Configuring the Module's Response to a Connection Failure on page 3-11).
- use the programming software to display fault data (See Displaying Fault Data on page 3-15)
- program logic to monitor fault data so you can take appropriate action (Refer to *Logix5000 Controllers Common Procedures Programming Manual*, publication number 1756-PM001B-EN-P, for examples.)

Displaying Fault Data

Fault data for certain types of module faults can be viewed through the programming software.

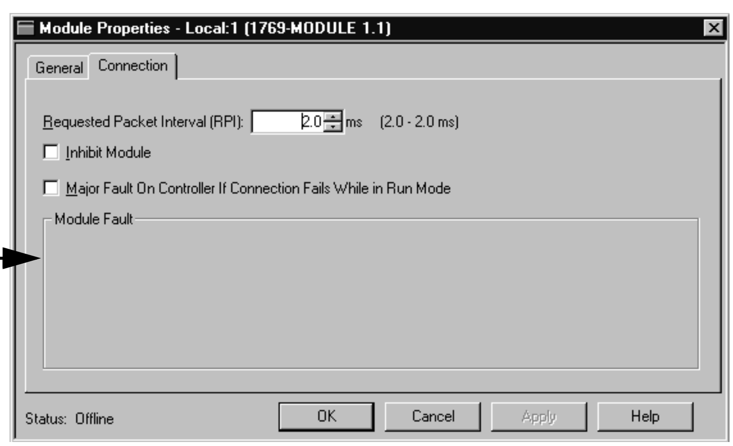
To view this data, select Controller Tags in the Controller Organizer. Right-click to select Monitor Tags.



If the module faults, but the connection to the controller remains open, the controller tags database displays the fault value 16#7000_0001. If the module is faulted and the connection to the controller is broken, the controller tags database displays a fault value of 16#FFFF_FFFF.

The display for the fault data defaults to decimal. Change it to Hex to read the fault code.

You can also view module fault data on the Connection tab of the Module Properties screen.



See your 1769 module's user documentation for a description of module faults. To recover from module faults, correct the module fault condition and send new data to the module by downloading the user program with configuration data, inhibiting and then uninhibiting the module, or cycling power.

End-Cap Detection and Module Faults

If a module that is not adjacent to an end cap experiences a fault and the connection to the controller is not broken, only the module enters its fault state.

If a module that is adjacent to an end cap experiences a fault, both the module and the controller transition to the fault state.

Communicating with Devices on a Serial Link

Using This Chapter

Default Communication Configuration

The CompactLogix5320 has the following default communication configuration.

Parameter	Default
Baud Rate	19.2K
Parity	none
Source ID (Node Address)	0
Control Lines	no handshaking
Error Detection	BCC
Embedded Responses	auto detect
Duplicate Packet (Message) Detect	enabled
ACK Timeout	50 counts
NAK retries	3 retries
ENQ retries	3 retries
Data Bits	8
Stop Bits	1
Protocol	DF1 point-to-point

NOTE Node Address is part of the default configuration. Changing the node address will result in DCH0 LED turning off.

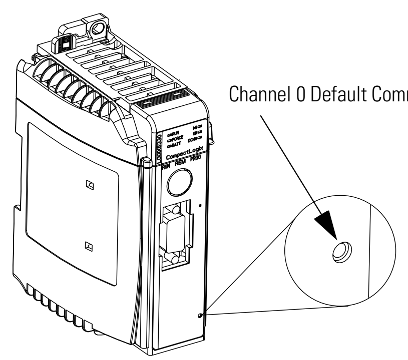
Using the Channel 0 Default Communication Push Button

Use the Channel 0 Default Communication Push Button to change from the user-defined communication configuration to the default communications configuration. Hold the button until the Channel 0 Default Communications (DCH0) LED turns on (green, steady) showing that the default communication configuration is active.

NOTE Before pressing the Channel 0 Default Communication Push Button, be sure you know the current communication configuration for Channel 0. If the channel is to be returned to this configuration, it must be entered manually while online with the controller or downloaded as part of a Logix Project file.

To accomplish this online, enter the Controller Properties screen under the Serial Port, System Protocol and User Protocol tabs.

The Channel 0 Default Communication Push Button is located on the front of the controller in the lower right corner.



NOTE The Default Communication Push Button is recessed.

Configuring Your System for a Serial Link

For the CompactLogix controller to operate on a serial network, you need:

- a workstation with a serial port
- RSLinx software to configure the serial communication driver
- RSLogix5000 programming software to configure the serial port of the controller

IMPORTANT	Limit the length of serial (RS-232) cables to 15.2m (50 ft.).

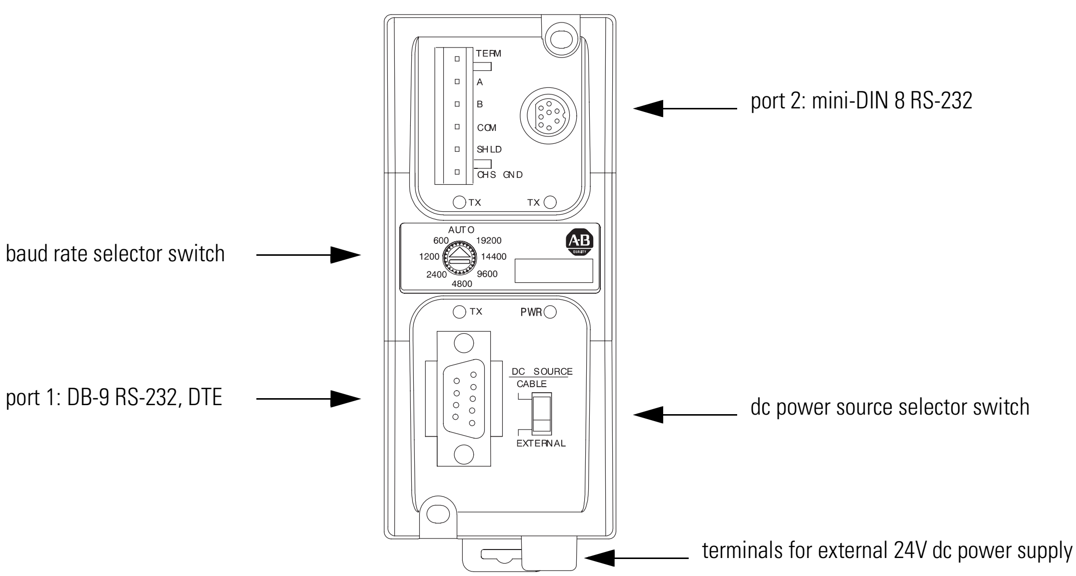
Step 1: Configure the Hardware

The RS-232 port is a non-isolated serial port built-in to the front of the CompactLogix controller.

1. Determine whether you need an isolator.

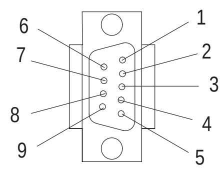
   If you connect the controller to a modem or an ASCII device, consider installing an isolator between the controller and modem or ASCII device. An isolator is also recommended when connecting the controller directly to a programming workstation.

   One possible isolator is the 1761-NET-AIC interface converter.

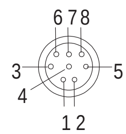


2. Select the appropriate cable.

<table>
<tr><th>If you are using an isolator:</th><th>Use this cable:</th></tr>
<tr><td>yes</td><td>
The 1761-CBL-AP00 cable (right-angle bend connector to controller) or the 1761-CBL-PM02 cable (straight connector to the controller) attaches the controller to port 2 on the 1761-NET-AIC isolator. The 8-pin mini-DIN connector is not commercially available, so you cannot make this cable.

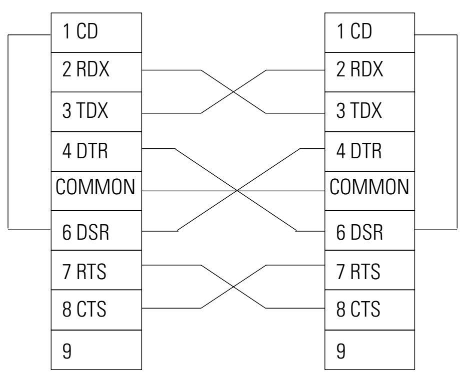


DB-9 right-angle or straight cable end



8-pin, mini-DIN cable end

<table>
<tr><th>Pin:</th><th>DB-9 end:</th><th>Mini-DIN end:</th></tr>
<tr><td>1</td><td>DCD</td><td>DCD</td></tr>
<tr><td>2</td><td>RxD</td><td>RxD</td></tr>
<tr><td>3</td><td>TxD</td><td>TxD</td></tr>
<tr><td>4</td><td>DTR</td><td>DTR</td></tr>
<tr><td>5</td><td>ground</td><td>ground</td></tr>
<tr><td>6</td><td>DSR</td><td>DSR</td></tr>
<tr><td>7</td><td>RTS</td><td>RTS</td></tr>
<tr><td>8</td><td>CTS</td><td>CTS</td></tr>
<tr><td>9</td><td>na</td><td>na</td></tr>
</table>
</td></tr>
<tr><td>no</td><td>
The 1756-CP3 cable attaches the controller directly to the RS-232 device.



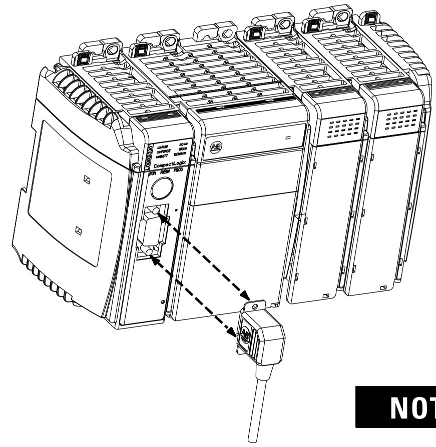
straight cable end

right-angle cable end

If you make your own cable, it must be shielded and the shields must be tied to the metal shell (that surrounds the pins) on both ends of the cable.

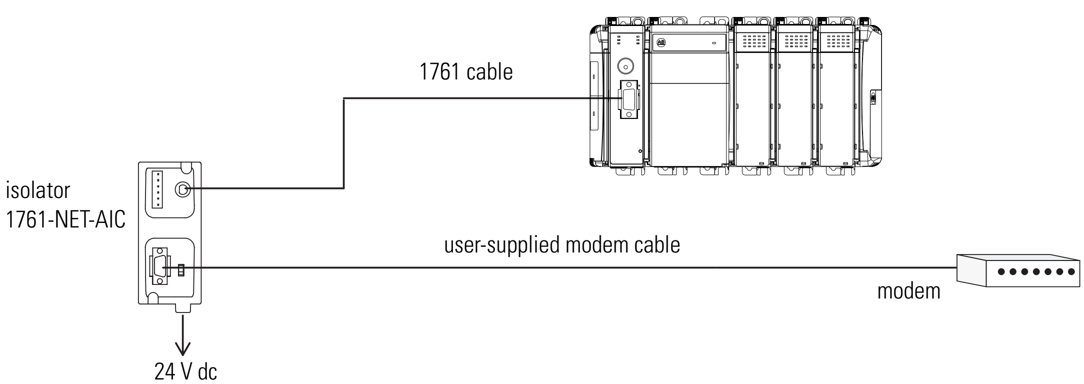
You can also use a 1747-CP3 cable from the SLC product family. This cable has a larger right-angle connector than the 1756-CP3 cable.
</td></tr>
</table>

3. Connect the appropriate cable to the serial port on the controller.



NOTE The RS-232 port is locally grounded.

4. If necessary, attach the controller to the isolator.

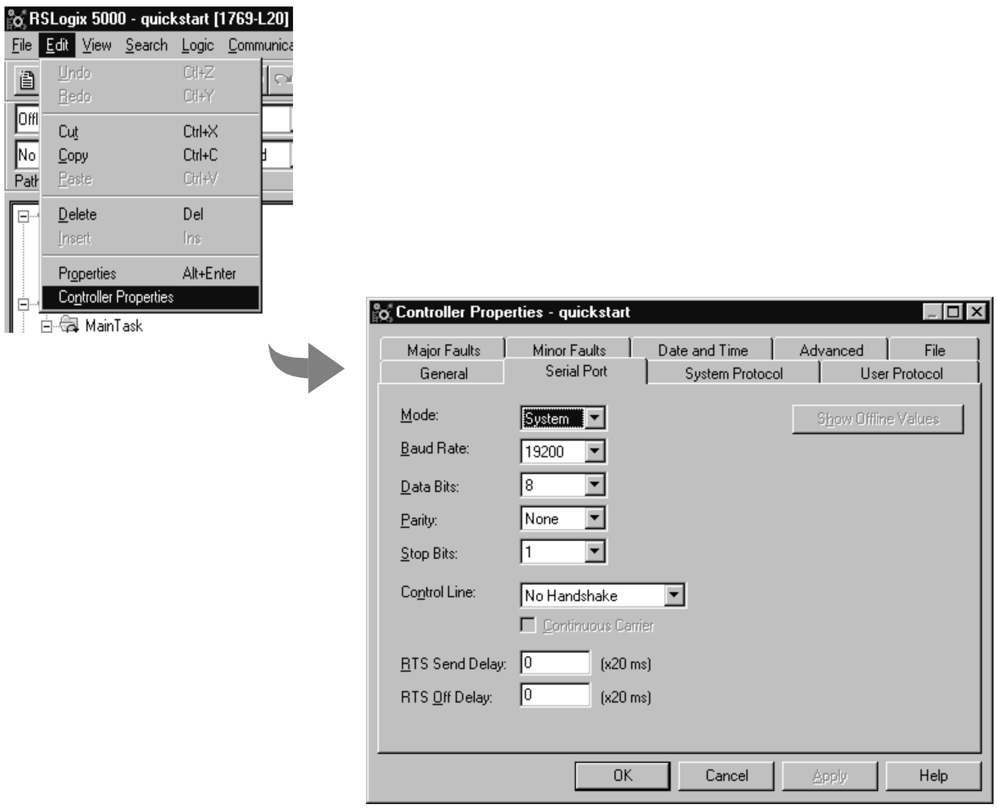


ATTENTION The CompactLogix controller is grounded through its DIN rail or mounting foot. It is important that you understand the workstation's grounding system before connecting it to the controller. An isolator is recommended between the controller and the workstation.

Step 2: Configure the Serial Port of the Controller

1. In RSLogix 5000 software, select Edit → Controller Properties.

2. On the Serial Port tab, specify the proper mode and the appropriate serial communication settings.



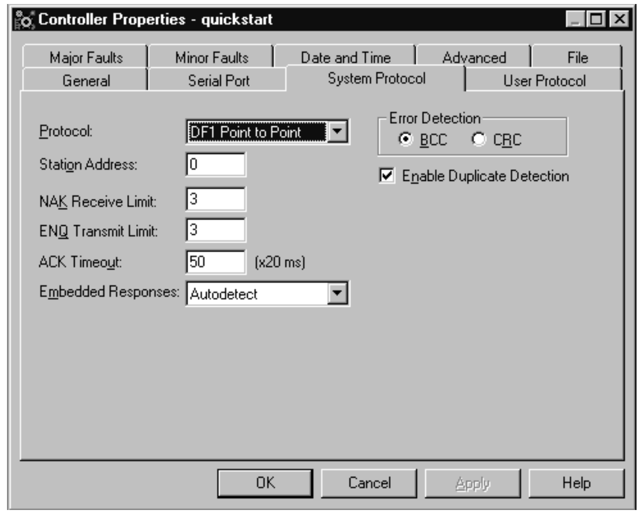
3. On the System Protocol tab, select DH-485 or the appropriate DF1 communication mode for point-to-point or master/slave communications. Or on the User Protocol tab, select ASCII to communicate with an ASCII device.

Specifying Serial Port Characteristics

Specify these characteristics on the Serial Port tab (default values are shown in bold):

Characteristic:	Description (default is shown in bold):
Mode	Select **System** (for DF1 and DH485 communication) or User mode (for ASCII communication).
Baud rate	Specifies the communication rate for the serial port. Select a baud rate that all devices in your system support. Select 110, 300 600, 1200, 2400, 4800, 9600, **19200**, 38400 Kbps. Note: 38400 Kbps only in DF1 mode
Parity	Specifies the parity setting for the serial port. Parity provides additional message-packet error detection. Select **None** or Even.
Data bits	Specifies the number of bits per message packet. Select **8**.
Stop bits	Specifies the number of stop bits to the device with which the controller is communicating. Select **1** or 2.
Control line	Specifies the mode in which the serial driver operates. Select **No Handshake**, Full-Duplex, Half-Duplex with Continuous Carrier, or Half-Duplex without Continuous Carrier. If you are not using a modem, select No Handshake. If both modems in a point-to-point link are full-duplex, select Full-Duplex for both controllers. If the master modem is full-duplex and the slave modem is half-duplex, select Full-Duplex for the master controller and select Half-Duplex with Continuous Carrier for the slave controller. If all the modems in the system are half-duplex, select Half-Duplex without Continuous Carrier for the controller.
RTS send delay	Enter a count that represents the number of 20 ms periods of time that elapse between the assertion of the RTS signal and the beginning of a message transmission. This time delay lets the modem prepare to transmit a message. The CTS signal must be high for the transmission to occur. The range is **0** to 32767 periods.
RTS off delay	Enter a count that represents the number of 20 ms periods of time that elapse between the end of a message transmission and the de-assertion of the RTS signal. This time delay is a buffer to make sure the modem successfully transmits the entire message. The range is **0** to 32767 periods. Normally leave at zero.

Specifying System Protocol Characteristics



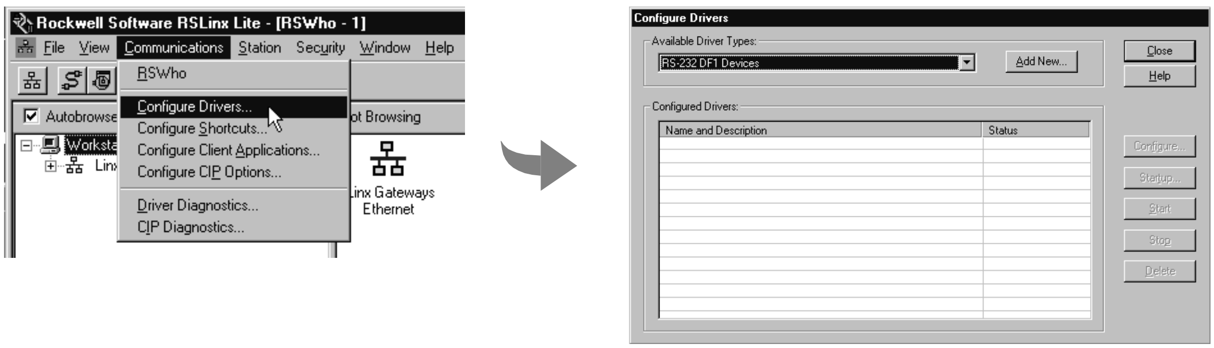
The available system modes are:

Use this mode:	For:	See page:
DF1 point-to-point	communication between the controller and one other DF1-protocol-compatible device. This is the default system mode. This mode is typically used to program the controller through its serial port.	4-10
DF1 master mode	control of polling and message transmission between the master and slave nodes. The master/slave network includes one controller configured as the master node and as many as 254 slave nodes. Link slave nodes using modems or line drivers. A master/slave network can have node numbers from 0 to 254. Each node must have a unique node address. Also, at least 2 nodes must exist to define your link as a network (1 master and 1 slave station are the two nodes).	4-13
DF1 slave mode	using a controller as a slave station in a master/slave serial communication network. When there are multiple slave stations on the network, link slave stations using modems or line drivers to the master. When you have a single slave station on the network, you do not need a modem to connect the slave station to the master; you can configure the control parameters for no handshaking. You can connect 2 to 255 nodes to a single link. In DF1 slave mode, a controller uses DF1 half-duplex protocol. One node is designated as the master and it controls who has access to the link. All the other nodes are slave stations and must wait for permission from the master before transmitting.	4-13
User mode	communicating with ASCII devices. This requires your program logic to use the ASCII instructions to read and write data from and to an ASCII device.	4-16
DH-485	communicating with other DH-485 devices multi-master, token passing network allowing programming and peer-to-peer messaging.	5-1

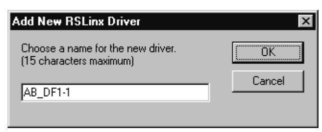
Step 3: Configure the Programming Terminal Communication Driver

Use RSLinx software to configure the serial communication driver. Select the “DF1” driver.

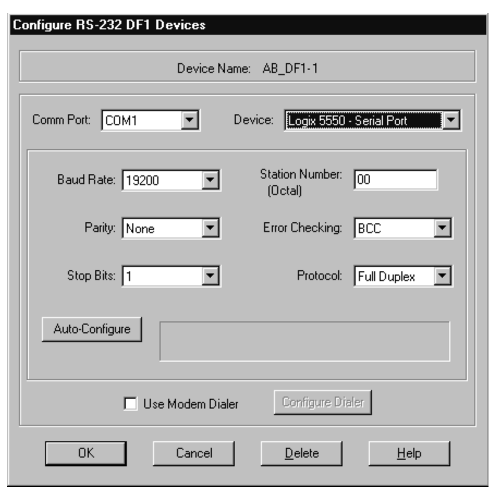
1. In RSLinx software, select Communication — Configure Driver. From the Available Driver Types list, select”RS-232 DF1 Devices“. Click Add New.



2. Specify a name for the driver.

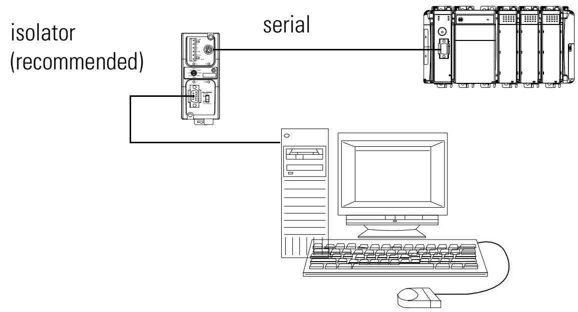


3. Specify the appropriate communication settings.Select the “Logix5550 serial port” as the Device and specify the COM port. Click Autoconfigure to have the software determine the remaining serial settings.



Example 1: Workstation Directly Connected to a CompactLogix Controller

In the following example, a workstation directly connects to a CompactLogix controller over a serial link. This is useful for downloading a controller project directly to the controller.



This type of protocol supports simultaneous transmission between two devices in both directions. The DF1 point-to-point protocol controls message flow, detects and signals errors, and retries if errors are detected.

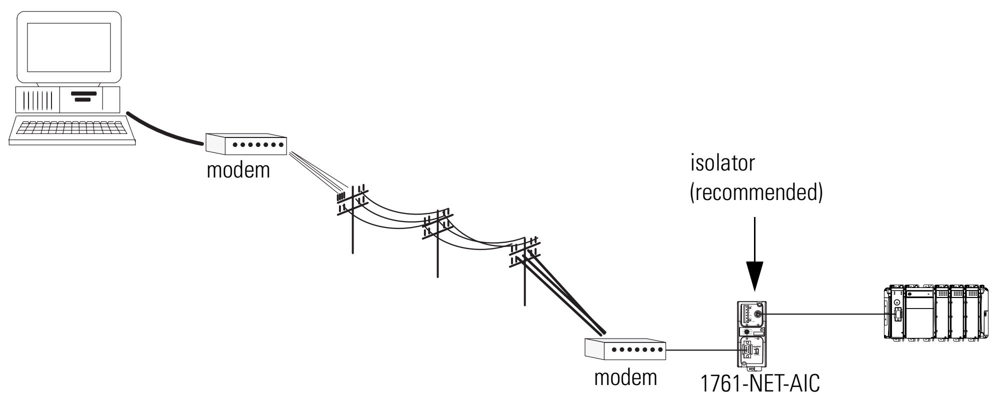
Configuring a DF1 Point-to-Point Station

This field:	Description:
Station address	The station address for the serial port on the DF1 point-to-point network. Enter a valid DF1 address (0 to 254). Address 255 is reserved for broadcast messages. The default is 0.
NAK receive limit	Specifies the number of NAKs the controller can receive in response to a message transmission. Enter a value 0 to 127. The default is 3.
ENQ transmit limit	Specifies the number of inquiries (ENQs) you want the controller to send after an ACK timeout. Enter a value 0 to 127. The default is 3.
ACK timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0 to 32767. Limits are defined in 20 ms intervals. The default is 50 (1000 ms).
Embedded response	Specifies how to enable embedded responses. Select Autodetect (enabled only after receiving one embedded response) or Enabled. The default is Autodetect.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

For information about communicating on DeviceNet with the 1761-NET-DNI, see Appendix F, *Communications on DeviceNet*.

Example 2: Workstation Remotely Connected to a CompactLogix Controller

In the following example, a workstation remotely connects to a CompactLogix controller over a serial link. A modem is connected to the controller to provide remote access.



If you use a modem to remotely connect the controller to one workstation, use DF1 point-to-point (full-duplex) protocol, as in the previous example.

Master/Slave Communication Methods

Half-duplex DF1 Protocol

Half-duplex master/slave protocol is a SCADA protocol, consisting of 1 master and up to 254 slaves. Typically, the master polls all of the slaves for data in a round-robin fashion, using RF modems, leased-line modems, or any similar media.

A master station can communicate with a slave station in two ways:

Name:	This method:	Benefits:
standard communication mode	initiates polling packets to slave stations according to their position in the polling array(s). Polling packets are formed based on the contents of the normal poll array and the priority poll array.	This communication method is most often used for point-to-multipoint configurations. This method provides these capabilities: • slave stations can send messages to the master station (polled report-by-exception) • slave stations can send messages to each other via the master (slave-to-slave transfers) • master maintains an active station array The poll array resides in a user-designated data file. You can configure the master: • to send messages during its turn in the poll array *or* • for between-station polls (master transmits any message that it needs to send before polling the next slave station) In either case, configure the master to receive multiple messages or a single message per scan from each slave station.
message-based communication mode	initiates communication to slave stations using only user-programmed message (MSG) instructions. Each request for data from a slave station must be programmed via a MSG instruction. The master polls the slave station for a reply to the message after waiting a user-configured period of time. The waiting period gives the slave station time to formulate a reply and prepare the reply for transmission. After all of the messages in the master's message-out queue are transmitted, the slave-to-slave queue is checked for messages to send.	If your application uses satellite transmission or public switched-telephone-network transmission, consider choosing message-based communication. Communication to a slave station can be initiated on an as-needed basis. Also choose this method if you need to communicate with non-intelligent remote terminal units (RTUs).

Configuring a DF1 Slave Station

This field:	Description:
Station address	The station address for the serial port on the DF1 slave. Enter a valid DF1 address (0 to 254). Address 255 is reserved for broadcast messages. The default is 0.
Transmit retries	The number of times the remote station retries a message after the first attempt before the station declares the message undeliverable. Enter a value 0 to 127. The default is 3.
Slave poll timeout	Specifies the amount of time the slave station waits to be polled by a master before indicating a fault. Enter a value 0 to 32767. Limits are defined in 20 ms intervals. The default is 3000 (60,000 ms).
EOT suppression	Select whether or not to suppress sending EOT packets in response to a poll. The default is not to suppress sending EOT packets.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

Configuring a DF1 Master Station

This field:	Description:
Station address	The station address for the serial port on the DF1 master. Enter a valid DF1 address (0 to 254). Address 255 is reserved for broadcast messages. The default is 0.
Transmit retries	Specifies the number of times a message is retried after the first attempt before being declared undeliverable. Enter a value 0 to 127. The default is 3.
ACK timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0 to 32767. Limits are defined in 20ms intervals. The default is 50 (1000 ms).
Reply message wait	**Message-based polling mode only** Specifies the amount of time the master station waits after receiving an ACK to a master-initiated message before polling the slave station for a reply. Enter a value 0 to 65535. Limits are defined in 20ms intervals. The default is 5 (100 ms).
Polling mode	Select one of these: • Message Based (slave cannot initiate messages) • Message Based (slave can initiate messages) - default • Standard (multiple message transfer per node scan) • Standard (single message transfer per node scan)
Master transmit	**Standard polling modes only** Select when the master station sends messages: • between station polls (default) • in polling sequence

This field:	Description:
Normal poll node tag	**Standard polling modes only** An integer tag array that contains the station addresses of the slave stations. Create a single-dimension array of data type INT that is large enough to hold all the normal station addresses. The minimum size is three elements. This tag must be controller-scoped. The format is: *list[0]* contains total number of stations to poll *list[1]* contains address of station currently being polled *list[2]* contains address of first slave station to poll *list[3]* contains address of second slave station to poll *list[n]* contains address of last slave station to poll
Normal poll group size	**Standard polling modes only** The number of stations the master station polls after polling all the stations in the priority poll array. Enter 0 (default) to poll the entire array.
Priority poll node tag	**Standard polling modes only** An integer tag array that contains the station addresses of the slave stations you need to poll more frequently. Create a single-dimension array of data type INT that is large enough to hold all the priority station addresses. The minimum size is three elements. This tag must be controller-scoped. The format is: *list[0]* contains total number of stations to be polled *list[1]* contains address of station currently being polled *list[2]* contains address of first slave station to poll *list[3]* contains address of second slave station to poll *list[n]* contains address of last slave station to poll
Active station tag	**Standard polling modes only** An array that stores a flag for each of the active stations on the DF1 link. Both the normal poll array and the priority poll array can have active and inactive stations. A station becomes inactive when it does not respond to the master's poll. Create a single-dimension array of data type SINT that has 32 elements (256 bits). This tag must be controller-scoped.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

If You Choose One of the Standard Polling Modes

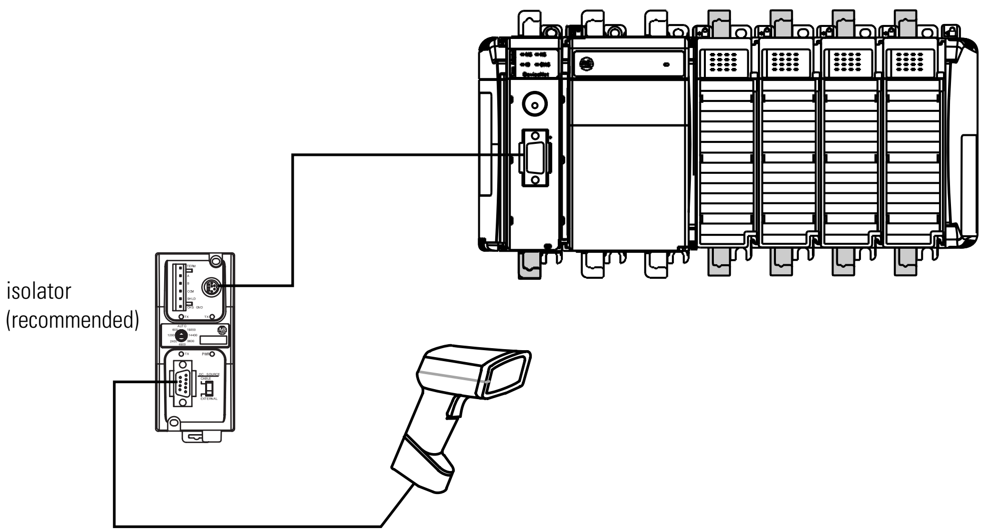
The master station polls the slave stations in this order:

1. all stations that are active in the priority poll array
2. one station that is inactive in the priority poll array
3. the specified number (normal poll group size) of active stations in the normal poll array
4. one inactive station, after all the active stations in the normal poll array have been polled

Use the programming software to change the display style of the active station array to binary so you can view which stations are active.

Example 3: CompactLogix Controller to a Bar Code Reader

In the following example, a workstation connects to a bar code reader. A bar code reader is an ASCII device, so you configure the serial port differently than in the previous examples. Configure the serial port for User mode, rather than the system mode.

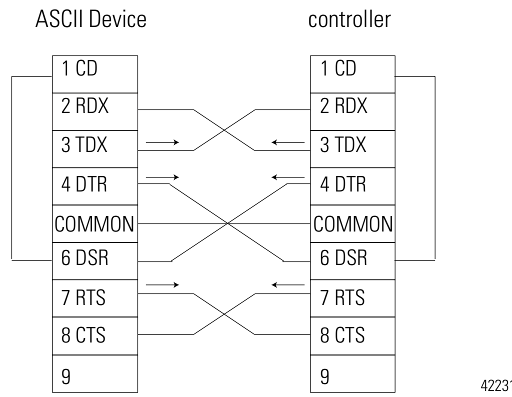
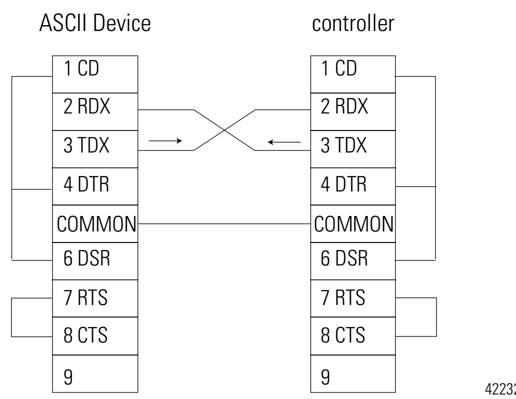


Connect the ASCII Device to the Controller

To connect the ASCII device to the serial port of the controller:

1. For the serial port of the ASCII device, determine which pins send signals and which pins receive signals.

2. Connect the sending pins to the corresponding receiving pins and attach jumpers:

<table>
<tr><th>If the communications hardware handshaking is:</th><th>Then wire the connectors as follows:</th></tr>
<tr><td>enabled</td><td>
</td></tr>
<tr><td>disabled</td><td>
</td></tr>
</table>

3. Attach the cable shield to both connectors and tie the cable to both connectors.

4. Connect the cable to the controller and the ASCII device.

The following table lists the default serial port configuration settings for the ASCII protocol. You specify these settings on the User Protocol tab under Controller Properties.

Configuring User Mode

This field:	Description:
Buffer size	Specify the maximum size (in bytes) of the data array you plan to send and receive. The default is 82 bytes.
Termination characters	Specify the characters you will use to designate the end of a line. The default characters are '$r' and '$FF'.
Append characters	Specify the characters you will append to the end of a line. The default characters are '$r' and '$l'.(1)
XON/XOFF	Select whether or not to regulate the flow of incoming data. The default is disabled.
Echo mode	Select whether or not to echo data back to the device from which it was sent. The default is disabled.
Delete mode	Select Ignore, CTR, or Printer for the delete mode. The default is Ignore.

(1) IEC 1131-3 representation for carriage return and line feed.

Programming ASCII Instructions

The controller supports ASCII instructions to communicate with ASCII devices. Your RSLogix5000 programming software CDROM includes programming examples using ASCII instructions.

For information about using these examples, see the *Logix5000 Controllers General Instruction Set Reference Manual*, publication 1756-RM001C-EN-P.

Communicating with Devices on a DH-485 Link

Using This Chapter

The DH-485 protocol uses RS-485 half-duplex as its physical interface. (RS-485 is a definition of electrical characteristics; it is *not* a protocol.) You can configure the RS-232 port of the CompactLogix controller to act as a DH-485 interface.

IMPORTANT A DH-485 network consists of multiple cable segments. Limit the total length of all the segments to 1219m (4000 ft.).

Configuring Your System for a DH-485 Link

For the CompactLogix controller to operate on a DH-485 network, you need:

- a 1761-NET-AIC interface converter for each CompactLogix controller you want to put on the DH-485 network.

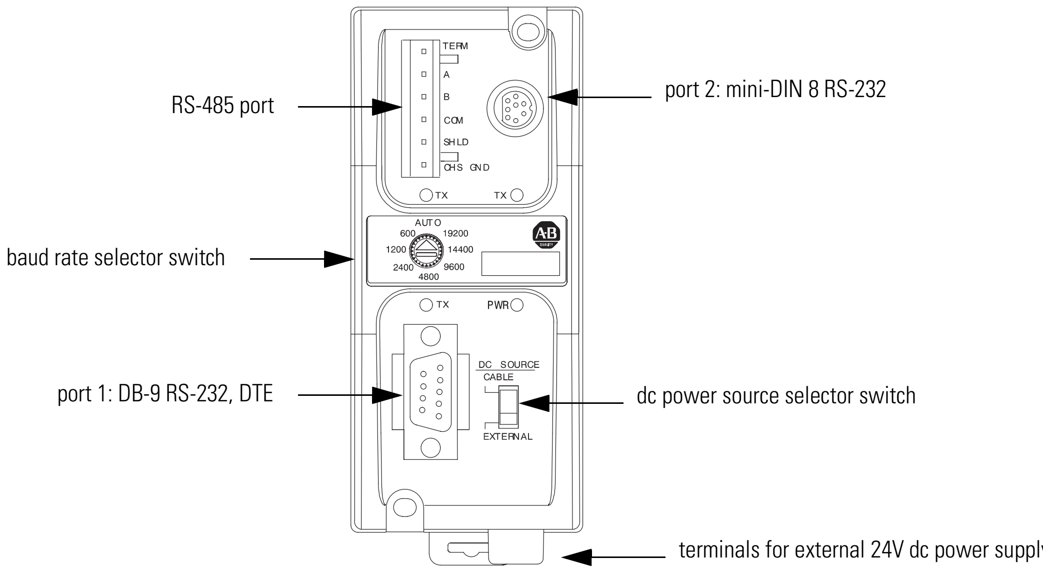
  You could have two controllers per one 1761-NET-AIC converter, but you would need a different cable for each controller. Connect one controller to port 1 (9-pin connector) and one controller to port 2 (mini-DIN connector).

- RSLogix 5000 programming software to configure the serial port of the controller for DH-485 communications.

Step 1: Configure the Hardware

The RS-232 port is a non-isolated serial port built-in to the front of the CompactLogix controller. The RS-232 port supports the requirements you need for the DH-485 network connection.

Connect the controller to an RS-232-to-RS-485 isolator. One possible isolator is the 1761-NET-AIC interface converter.



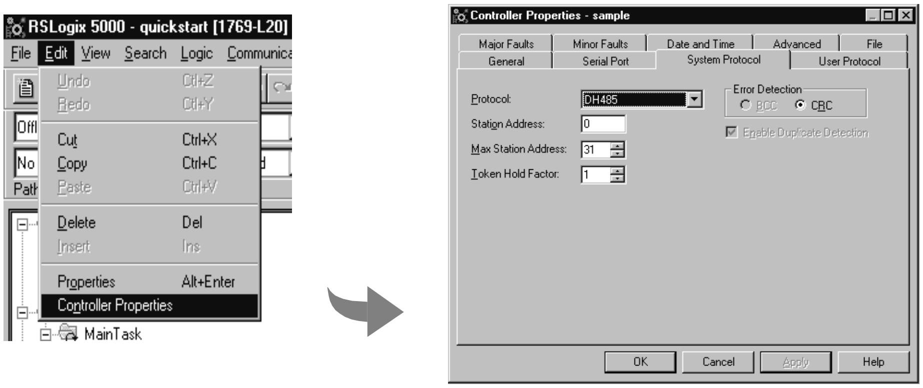
Connect the serial port of the CompactLogix controller to either port 1 or port 2 of the 1761-NET-AIC converter. Use the RS-485 port to connect the converter to the DH-485 network.

The cable you use to connect the controller depends on the port you use on the 1761-NET-AIC converter.

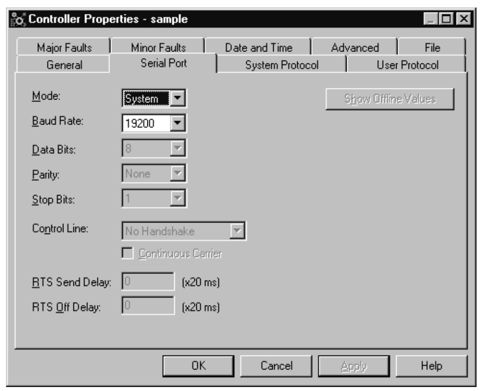
If you connect to this port:	Use this cable:
port 1 DB-9 RS-232, DTE connection	1747-CP3 or 1761-CBL-AC00
port 2 mini-DIN 8 RS-232 connection	1761-CBL-AP00 or 1761-CBL-PM02

Step 2: Configure the DH-485 Port of the Controller

1. In RSLogix 5000 software, select Edit — Controller Properties. On the System Protocol tab, select DH-485.



2. On the Serial Port tab, specify the appropriate communication settings.



The grayed out settings are selections that do not apply to a DH-485 network.

Specify these characteristics on the Serial Port tab (default values are shown in bold):

Characteristic:	**Description (default is shown in bold):**
Baud Rate	Specifies the communication rate for the DH-485 port. All devices on the same DH-485 network must be configured for the same baud rate. Select 9600 or **19200** Kbps.
Node Address	Specifies the node address of the CompactLogix controller on the DH-485 network. Select a number **1**-31 decimal, inclusive. To optimize network performance, assign node addresses in sequential order. Masters, such as personal computers, should be assigned the lowest address numbers to minimize the time required to initialize the network.
Token Hold Factor	Specifies the node address of the CompactLogix controller on the DH-485 network. Select a number **1**-4, inclusive.
Maximum Node Address	Specifies the maximum node address of all the devices on the DH-485 network. Select a number 1-**31** decimal, inclusive. To optimize network performance, make sure: • the maximum node address is the highest node number being used on the network • that all the devices on the same DH-485 network have the same selection for the maximum node address.

Planning a DH-485 Network

The DH-485 network offers:

- interconnection of 32 devices
- multi-master capability
- token passing access control
- the ability to add or remove nodes without disrupting the network
- maximum network length of 1219 m (4000 ft.)

The DH-485 protocol supports two classes of devices: masters and slaves. All masters on the network get a chance to initiate message transfers. The DH-485 protocol uses a token-pass algorithm to determine which master has the right to transmit.

DH-485 Token Rotation

A node holding the token can send any valid packet onto the network. As a default, each node gets only one transmission (plus two retries) each time it receives the token. After a node sends one message packet, it attempts to give the token to its successor by sending a "token pass" packet to its successor.

If no network activity occurs, the master sends the token pass packet again. After two retries (a total of three tries) the master attempts to find a new successor.

IMPORTANT The maximum address that the master searches for before starting again with zero is the value in the configurable parameter "maximum node address." The default and maximum value for this parameter is 31 for all masters and slaves.

The allowable range of the node address of a master is 0 to 31. The allowable address range for all slaves is 1 to 31. There must be at least one master on the network.

Network Initialization

The network requires at least one master to initialize it. Network initialization begins when a master on the network detects a period of inactivity that exceeds the time of a link dead timeout. When the link dead timeout is exceeded, usually the master with the lowest address claims the token. When a master has the token it will begin to build the network.

Building a network begins when the master that claimed the token tries to pass the token to the successor node. If the attempt to pass the token fails, or if the master has no established successor (for example, when it powers up), it begins a linear search for a successor starting with the node above it in the addressing.

When the master finds another active node, it passes the token to that node, which repeats the process until the token is passed all the way around the network to the initial node. At this point, the network is in a state of normal operation.

Number of Nodes and Node Addresses

The number of nodes on the network directly affects the data transfer time between nodes. Unnecessary nodes (such as a second programming terminal that is not being used) slow the data transfer rate. The maximum number of nodes on the network is 32.

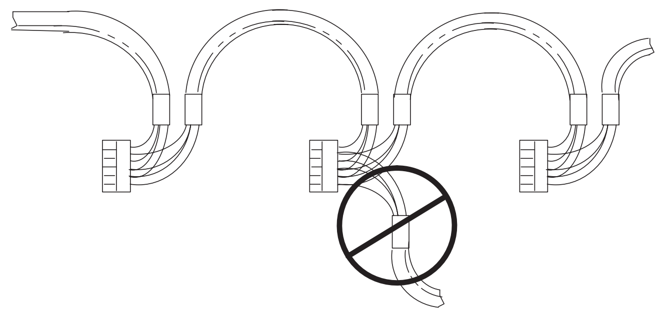
If the node addresses for controllers are assigned in sequence, starting at node 1 (with node 0 left for a programming terminal), it is as efficient to leave the maximum node address at 31 as it is to decrease it to the highest node address on the network. Then, adding devices to the network at a later time will not require modifying the maximum node address in every device on the network. The maximum node address should be the same for all devices on a DH-485 network for optimal operation.

The best network performance occurs when node addresses start at 0 and are assigned in sequential order. The controller defaults to node address 1. Masters, such as personal computers, should be assigned the lowest numbered addresses to minimize the time required to initialize the network.

Installing a DH-485 Network

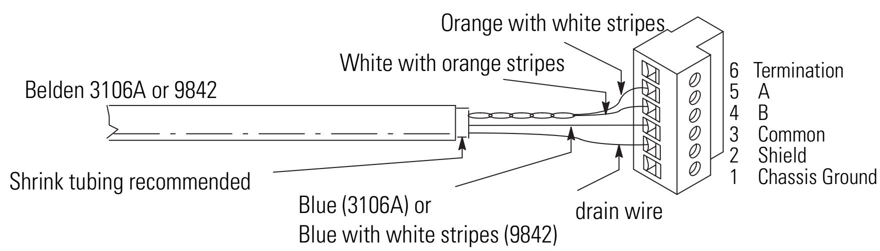
A DH-485 network consists of a number of cable segments daisy-chained together. The total length of the cable segments cannot exceed 1219 m (4000 ft).

IMPORTANT Use shielded, twisted-pair cable - either Belden 3106A or Belden 9842. A daisy-chained network is recommended.

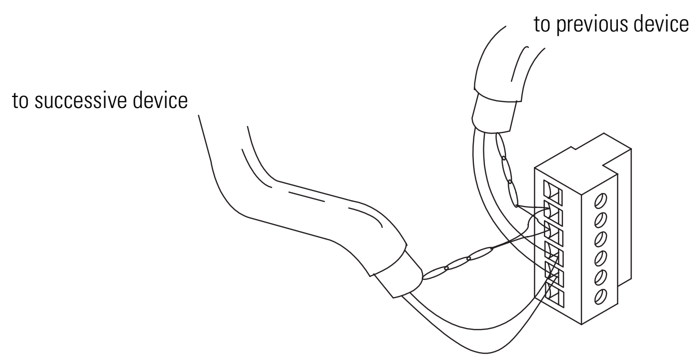


When cutting cable segments, make them long enough to route them from one link coupler to the next with sufficient slack to prevent strain on the connector. Allow enough extra cable to prevent chafing and kinking in the cable.

Single Cable Connection



Multiple Cable Connection



The table below shows wire/terminal connections for Belden 3106A.

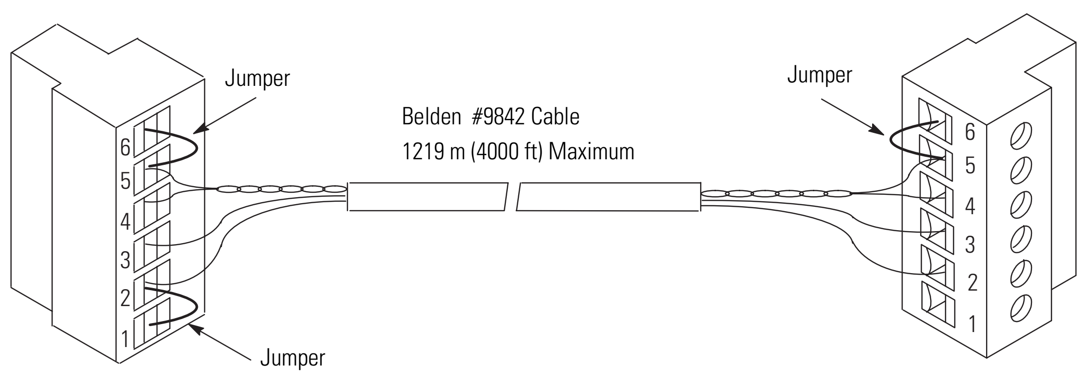
For this Wire/Pair	Connect this Wire	To this Terminal
shield/drain	non-jacketed	2 - Shield
blue	blue	3 - (Common)
white/orange	white with orange stripe	4 - (Data B)
	orange with white stripe	5 - (Data A)

The table below shows wire/terminal connections for Belden 9842.

For this Wire/Pair	Connect this Wire	To this Terminal
shield/drain	non-jacketed	2 - Shield
blue/white	white with blue stripe	cut back - no connection(1)
	blue with white stripe	3 - (Common)
white/orange	white with orange stripe	4 - (Data B)
	orange with white stripe	5 - (Data A)

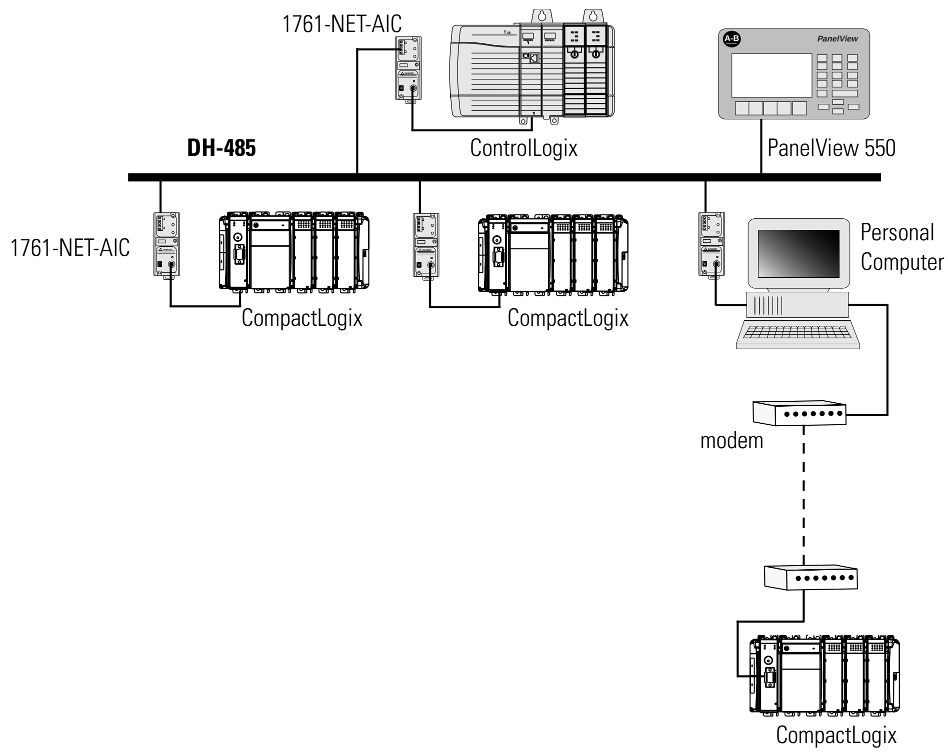
(1) To prevent confusion when installing the communication cable, cut back the white with blue stripe wire immediately after the insulation jacket is removed. This wire is not used by DH-485.

Grounding and Terminating a DH-485 Network



Example: CompactLogix Controller on a DH-485 Network

In the following example, both a CompactLogix controller and a ControlLogix controller use 1761-NET-AIC converter to connect to a DH-485 network.



CompactLogix System Specifications

Using This Appendix

CompactLogix Controller

Description	Value
Backplane Current	600 mA at +5V dc 0 mA at +24V dc
Operating Temperature	0° to +60°C (+32° to +140°F)
Storage Temperature	-40° to +85°C (-40° to +185°F)
Relative Humidity	5% to 95% non-condensing
Vibration	Operating: 10 to 500 Hz, 5G, 0.030 mm (in.) peak-to-peak
Shock	Operating: 30G, 11 ms, panel mounted (20G, 11 ms, DIN rail mounted) Non-operating: 40G, panel mounted (30G, DIN rail mounted)
Power Supply Distance Rating	4 (The controller must be within 4 slot positions of the power supply.)
Shipping Weight	0.325 kg (0.715 lbs)
Battery	1747-BA
Programming Cable	1747-CP3 or 1756-CP3
Agency Certification	• C-UL certified (under CSA C22.2 No.142) • UL 508 listed • CE compliant for all applicable directives
Hazardous Environment Class	Class I, Division 2, Hazardous Location, Groups A, B, C, D (UL 1604, C-UL under CSA C22.2 No. 213)
Radiated and Conducted Emissions	EN50081-2 Class A
Electrical /EMC:	*The unit has passed testing at the following levels:*
ESD Immunity (IEC1000-4-2)	4 kV contact, 8 kV air, 4 kV indirect
Radiated Immunity (IEC1000-4-3)	10 V/m, 80 to 1000 MHz, 80% amplitude modulation, +900 MHz keyed carrier
Fast Transient Burst (IEC1000-4-4)	2 kV, 5 kHz
Surge Immunity (IEC1000-4-5)	1 kV galvanic gun
Conducted Immunity (IEC1000-4-6)	10V, 0.15 to 80 MHz(1)

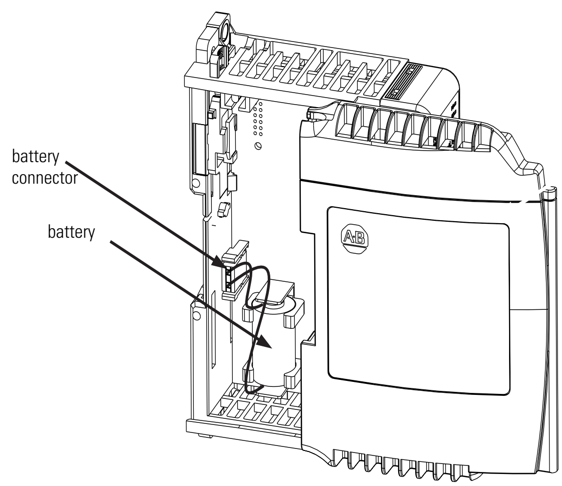
(1) Conducted Immunity frequency range may be 150 kHz to 30 MHz if the Radiated Immunity frequency range is 30 MHz to 1000 MHz.

IMPORTANT The amount of memory that the software displays includes both the user available memory and the memory reserved for overhead. Certain operations dynamically allocate and de-allocate user-available memory. See the specifications for your controller and Appendix E to estimate how much memory you have available for programming.

1747-BA Battery

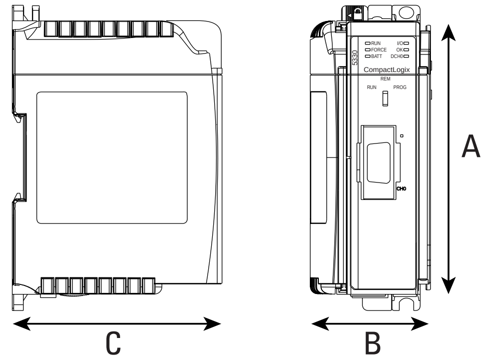
The CompactLogix controller uses the 1747-BA battery:

Battery	1747-BA containing 0.59g lithium



Dimensions

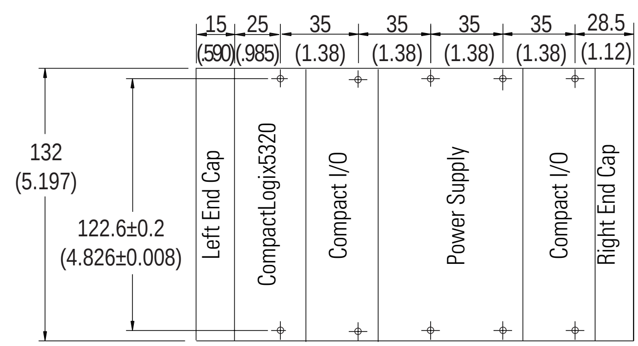
CompactLogix Modular Controller



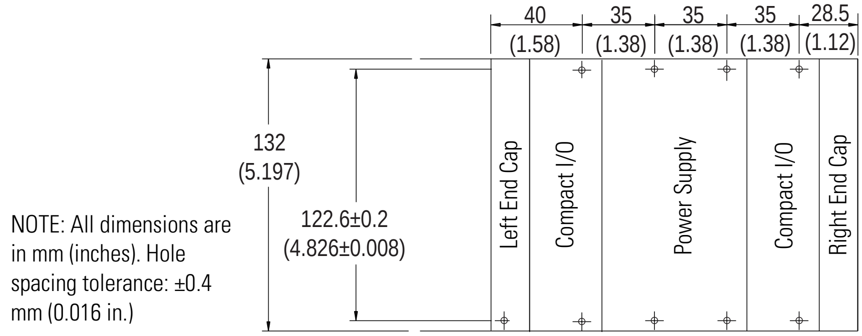
Dimension	1769-L20 CompactLogix5320
Height (A)	118 mm (4.649 in.)
Width (B)	50 mm (1.97 in.)
Depth (C)	87 mm (3.43 in.)

CompactLogix System

NOTE: All dimensions are in mm (inches). Hole spacing tolerance: ±0.4 mm (0.016 in.)



Compact I/O Expansion Power Supply and End Caps



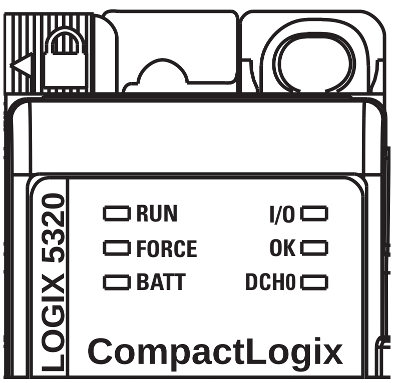
NOTE Compact I/O expansion cables have the same dimensions as the end caps.

CompactLogix Troubleshooting

Using This Appendix

CompactLogix Controller LEDs

The controller status LEDs provide a mechanism to determine the current status of the controller if a programming device is not present or available.



Indicator	Color/Status	Description
RUN	Off	no task(s) running; controller in Program mode
	Green	one or more tasks are running; controller is in the Run mode
FORCE	Off	no forces enabled
	Amber	forces enabled
	Amber Flashing	one or more input or output addresses have been forced to an On or Off state, but the forces have not been enabled
BAT	Off	battery supports memory
	Red	battery may not support memory, replace battery
OK	Off	no power applied
	Green	controller OK
	Red flashing	recoverable controller fault
	Red	non-recoverable controller fault • clear faults • clear memory • replace the controller
I/O	Off(1)	no activity; no I/O or communications configured
	Green	communicating to all devices
	Green flashing	one or more devices not responding
	Red flashing	not communicating to any devices controller faulted
DCH0	Off	user-configured communications active
	Green	default communications active
Channel 0	Off	no activity
	Green flickering	data is being received or transmitted

(1) If the controller does not contain an application (controller memory is empty), the I/O indicator will be off.

Identifying Controller Faults

Refer to the *Logix5000 Controllers Common Procedures Manual,* publication number 1756-PM001B-EN-P, for a list of controller fault messages that can occur during operation of the CompactLogix controller. Each description includes the error code, the probable cause, and the recommended corrective action.

The *Logix5000 Controllers Common Procedures Manual* also contains procedures for monitoring faults and developing fault routines.

Calling Rockwell Automation for Assistance

If you need to contact Rockwell Automation or local distributor for assistance, it is helpful to obtain the following (prior to calling):

- controller type, series letter, and revision letter of the unit
- series letter, revision letter, and firmware (FRN) number of the controller (as reported by the software)
- controller LED status
- controller error codes

Maintaining the Battery

Using this Appendix

For information about:	See page
Storing replacement batteries	C-1
Estimating battery life	C-1
Replacing batteries	C-3

Storing Replacement Batteries

Because a battery may leak potentially dangerous chemicals if stored improperly, store batteries as follows:

ATTENTION

Store batteries in a cool, dry environment. We recommend 25°C with 40% to 60% relative humidity. You may store batteries for up to 30 days between -45° to 85°C, such as during transportation. To avoid possible leakage, *do not* store batteries above 60°C for more than 30 days.

Estimating Battery Life

When the battery is about 95 percent discharged, the processor provides the following warnings:

- On the front of the processor, the BATTERY LED turns on (solid red).
- A minor fault occurs (type 10, code 10).

To estimate how long the battery will support the memory of the processor:

1. Determine the temperature (°C) 1 inch below the CompactLogix processor.

2. Determine the percentage of time that the processor is powered off per week.

EXAMPLE

If a processor is off:

- 8 hr/day during a 5-day work week
- all day Saturday and Sunday

Then the processor is off 52% of the time:

1. total hours per week = 7 x 24 = 168 hours
2. total off hours per week = (5 days x 8 hrs/day) + Saturday + Sunday = 88 hours
3. percentage off time = 88/168 = 52%

Use the off-time percentage you calculated with the following table to determine battery life:

Temperature:	Worst-case battery life estimate: Power off 100%:	Battery duration after the LED turns on:(1)
60°C	12 months	9 days
25°C	18 months	14 days

(1) The battery indicator (BATTERY) warns you when the battery is low. These durations are the amounts of time the battery will retain processor memory from the time the processor is powered down after the LED first turns on.

IMPORTANT

If the BATTERY LED turns on when you apply power to the processor, the battery life may be less than the table above indicates. Some of the warning time may have been used while the processor was off and unable to turn on the BATTERY LED.

Replacing a Battery

Because the processor uses a lithium battery, you must follow specific precautions when handling or disposing a battery.

ATTENTION The processor uses a lithium battery, which contains potentially dangerous chemicals. Before handling or disposing of a battery, review *Guidelines for Handling Lithium Batteries*, publication AG-5.4.

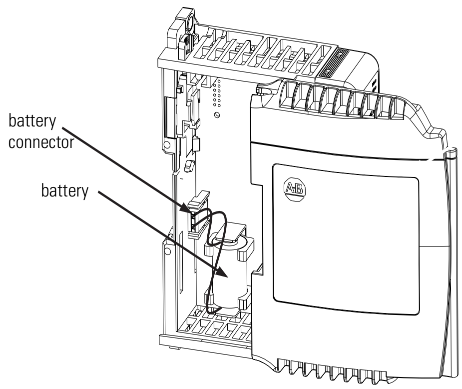
Follow the procedure below to replace the battery.

ATTENTION The user program will be lost when the battery is removed. Make a copy of your user program before removing and replacing the battery.

1. Save your user program.
2. Turn off power to the CompactLogix processor.
3. Does the existing battery show signs of leakage or damage?

If:	Then:
Yes	Before handling the battery, review *Guidelines for Handling Lithium Batteries*, publication AG-5.4.
No	Go to the next step.

4. Remove the old battery.



5. Install a new 1747-BA battery.

ATTENTION	Only install a 1747-BA battery. If you install a different battery, you may damage the processor.

6. Write battery date on door of processor.

7. Apply power to the processor. On the front of the processor, is the BATTERY LED off?

If:	Then:
Yes	Go to the next step.
No	Remove power, then: A. Check that the battery is correctly connected to the processor. B. If the BATTERY LED remains on, install another 1747-BA battery. C. If the BATTERY LED remains on after you complete Step B., contact your Rockwell Automation representative or local distributor.

8. Download your user program.

9. Dispose the old battery according to state and local regulations.

ATTENTION	Do not incinerate or dispose lithium batteries in general trash collection. They may explode or rupture violently. Follow state and local regulations for disposal of these materials. You are legally responsible for hazards created while your battery is being disposed.

Execution Time

Using This Appendix

Use this appendix to calculate the time required for the execution of an instruction. Times are for the CompactLogix5320 controller (1769-L20) and RSLogix 5000 programming software, version 7.0.

To estimate the execution time of a:	See page:
ladder instruction	D-1
function block routine	D-12

IMPORTANT Actual execution times may vary because of the configuration of your project and the version of RSLogix 5000 software that you are using.

IMPORTANT These execution times were measured with the CompactBus inhibited.

Estimate the Execution Time of a Ladder Instruction

To calculate the execution time of an instruction:

1. Use Table D.2 on page D-3 to find the base time for an instruction.

2. To the base time of the instruction, add time for expressions, subscripts, and data conversions:

If an operand:	Then add time for:	Using this table:
contains an expression	each operation in the expression. For each operation, use the time for the corresponding instruction.	Table D.2 on page D-3
is an element of an array and one or more of the subscripts is a tag (e.g., *tag_c[tag_d]*)	the calculation of the array subscript	Table D.5 on page D-11
is converted by sign-extension	the data conversion	Table D.1 on page D-3
contains a SINT or INT tag in an array subscript	the conversion of the SINT or INT to a DINT	

Examples:

Instruction:	Data Type:	Execution Time:
CMP (*tag_a * tag_b / 100)*		4.6 µs overhead
* (MUL)	REAL	19.1 µs for the MUL operation
/ (DIV)	REAL	19.9 µs for the DIV operation
		43.6 µs total
ADD		11.4 µs for the ADD instruction
Source A	DINT	9.0 µs for DINT to REAL conversion
Source B	REAL	
Destination	REAL	
		20.4 µs total
ADD		11.4 µs for the ADD instruction
Source A	DINT[0,DINT]	25.5 µs for array subscript 9.0 µs for DINT to REAL conversion of array element
Source B	REAL	
Destination	REAL	
		45.9 µs total
MUL		19.1 µs for MUL instruction
Source A	REAL	
Source B	DINT	9.0 µs for DINT to REAL conversion
Destination	DINT	13.2 µs for REAL to DINT conversion
		41.3 µs total

Reference Tables

Table D.1 Sign-Extension Conversion Times

To convert a:	To a:	Requires (μs):
SINT	DINT	1.1
	REAL	2.2
INT	DINT	1.2
	REAL	2.4
DINT	SINT	7.9
	INT	8.0
	REAL	9.0
REAL	SINT	15.9
	INT	16.0
	DINT	13.2

Table D.2 Instruction Execution Times

Instruction:	Optimal Data Type:	True Time (μs):	False Time (μs):	Notes:
ABL	na	13.6	11.8	
ABS	DINT	0.65	0.17	
	REAL	0.95	0.19	
ACB	na	13.3	11.7	
ACL	na	127.0	0.06	
ACS	REAL	289.3	0.12	
ADD	DINT	0.60	0.10	
	REAL	11.4	0.11	
AFI	na	0.06		
AHL	na	13.7	12.1	
AND	DINT	0.64	0.11	
ARD	na	79.0	15.5	
ARL	na	79.2	15.5	
ASN	REAL	279.3	0.11	
ATN	REAL	238.6	0.11	
AVE	REAL	46.8 + (Length * 15.6)	23.5	
AWA	na	82.6	15.3	
AWT	na	81.7	15.3	
BRK	na	0		

Table D.2 Instruction Execution Times (Continued)

Instruction:	Optimal Data Type:	True Time (μs):	False Time (μs):	Notes:
BSL	DINT	8.4 + ((Length/32) * 0.83)	4.3	Round up Length/32 to a whole number.
BSR	DINT	9.3 + ((Length/32) * 0.83)	4.3	Round up Length/32 to a whole number.
BTD	DINT	10.61	0.16	
CLR	DINT	0.37	0.10	
	REAL	0.43	0.11	
CMP	Refer to the operations within the expression.	4.6 + *expression*	0.11	
COP	SINT	7.4 + (Length * 0.09)		
	INT	7.1 + (Length * 0.19)		
	DINT	7.5 + (Length * 0.37)	0.15	
	REAL	7.6 + (Length * 0.37)	0.15	
COS	REAL	236.7	0.08	
CPT	Refer to the operations within the expression.	5.1 + *expression*	0.11	
CTD	COUNTER	0.46	0.43	
CTU	COUNTER	0.46	0.43	
DDT	DINT		9.1	Based on All mode
0 mismatches		29.6 + (Cmp. Length * 1.5)		
1 mismatch		41.5 + (Cmp. Length * 1.5)		
2 mismatches		44.2 + (Cmp. Length * 1.5)		
DEG	REAL	20.0	0.11	
DIV	DINT	11.7	0.10	
	REAL	19.9	0.11	
DTR	DINT	3.02	2.8	
EQU	DINT	0.40	0.10	
	REAL	0.40	0.11	
FAL	Refer to the operations within the expression.	10.9 + (*n* * (4.3 + *expression*)	7.3	*n* is the number of elements manipulated in one scan.
FBC	DINT		9.2	
0 mismatches		14.0 + (Cmp. Length * 1.5)		Based on All mode
1 mismatch		13.9 + (Cmp. Length * 1.5)		
2 mismatches		28.3 + (Cmp. Length * 1.5)		

Table D.2 Instruction Execution Times (Continued)

Instruction:	Optimal Data Type:	True Time (μs):	False Time (μs):	Notes:
FFL	SINT	10.4		
	INT	11.6		
	DINT, REAL	10.8	7.5	
FFU	SINT	11.8 + (Length * 0.66)		
	INT	12.9 + (Length * 1.11)		
	DINT, REAL	13.3 + (Length * 0.60)	7.7	
FLL	SINT	4.7 + (Length * 0.31)		
	INT	5.0 + (Length * 0.31)		
	DINT, REAL	5.6 + (Length * 0.26)	0.16	
FOR	DINT	16.6 + ((Terminal value/Step size) * 7.8)	3.0	
FRD	na	9.6	0.10	
FSC	Refer to the operations within the expression.	11.4 + (*n* * (4.2 + *expression*)	7.8	*n* is the number of elements manipulated in one scan.
GEQ	DINT	0.40	0.11	
	REAL	0.63	0.11	
GRT	DINT	0.40	0.11	
	REAL	0.64	0.11	
GSV	na	See Table D.3 on page D-8	0.16	
JMP	na	1.6	0.10	
JSR	na	11.4		No parameters
JSR/SBR	SINT	23.0 + (*number of parameters* * 3.8)		The time is for the JSR/SBR pair.
	INT	22.7 + (*number of parameters* * 4.2)		
	DINT	22.8 + (*number of parameters* * 3.5)		
	REAL	22.6 + (*number of parameters* * 3.6)		
JSR/RET	SINT	22.2 + (*number of parameters* * 3.8)		The time is for the JSR/RET pair.
	INT	21.8 + (*number of parameters* * 4.2)		
	DINT	22.0 + (*number of parameters* * 3.5)		
	REAL	21.7 + (*number of parameters* * 3.6)		
LBL	na	0.26		
LEQ	DINT	0.40	0.11	
	REAL	0.63	0.11	
LES	DINT	0.40	0.10	
	REAL	0.63	0.11	

Table D.2 Instruction Execution Times (Continued)

Instruction:	Optimal Data Type:	True Time (μs):	False Time (μs):	Notes:
LFL	SINT	10.4		
	INT	11.6		
	DINT	10.8	7.5	
	REAL	10.9	7.5	
LFU	SINT	12.9		
	INT	14.3		
	DINT, REAL	15.1	7.6	
LIM	DINT	0.85	0.11	
	REAL	3.8	0.11	
LN	REAL	204.8	0.10	
LOG	REAL	205.3	0.11	
MCR	na	0.05	0.05	
MEQ	DINT	0.63	0.10	
MOD	DINT	21.6	0.22	
	REAL	68.1	0.20	
MOV	DINT	0.48	0.11	
	REAL	0.88	0.10	
MSG	MESSAGE	98.0	7.0	
MUL	DINT	10.2	0.11	
	REAL	19.1	0.11	
MVM	DINT	9.5	0.16	
NEG	DINT	0.60	0.11	
	REAL	1.0	0.11	
NEQ	DINT	0.40	0.10	
	REAL	0.41	0.11	
NOP	na	0.05	0.06	
NOT	DINT	0.57	0.11	
ONS	BOOL	2.97	2.7	
OR	DINT	0.64	0.11	
OSF	BOOL	3.63	4.0	
OSR	BOOL	3.91	3.5	
OTE	BOOL	0.20	0.19	
OTL	BOOL	0.20	0.05	
OTU	BOOL	0.20	0.05	

Table D.2 Instruction Execution Times (Continued)

Instruction:	Optimal Data Type:	True Time (μs):	False Time (μs):	Notes:
PID	PID		18.4	
Independent		378.4		
ISA-dependent		451.7		
Manual mode		330.2		
Set output mode		330.2		
Independent- slave mode		402.0		
RAD	REAL	19.6	0.10	
RES	CONTROL, COUNTER, or TIMER	0.34	0.21	
RET (in FOR loop)	na	4.9		
RTO	TIMER	0.46	0.31	
SIN	REAL	243.3	0.09	
SQI	DINT	3.8	0.16	
SQL	DINT	6.6	3.9	
SQO	DINT	6.6	3.8	
SQR	DINT	10.5	0.10	
	REAL	39.8	0.11	
SRT	DINT	32.4 + x	4.6	Time "x" varies with the length and randomness of the numbers.
	REAL	33.3 + x	4.3	
SSV	na	See Table D.4 on page D-10	0.15	
STD	SINT	113.3 + (Length * 55.1)	22.0	
	INT	120.2 + (Length * 55.7)	25.3	
	DINT	120.3 + (Length * 54.6)	25.3	
	REAL	122.2 + (Length * 59.2)	25.4	
SUB	DINT	0.60	0.10	
	REAL	11.3	0.11	
TAN	REAL	307.7	0.08	
TND	na	0.01	0.11	
TOD	na	15.9	0.10	
TOF	TIMER	0.34	0.42	
TON	TIMER	0.46	0.34	
TRN	DINT	13.9	0.21	
	REAL	22.5	0.21	
UID	na	35.3	2.6	

Table D.2 Instruction Execution Times (Continued)

Instruction:	Optimal Data Type:	True Time (μs):	False Time (μs):	Notes:
UIE	na	38.0	2.6	
XIC	BOOL	0.11	0.05	
XIO	BOOL	0.12	0.05	
XOR	DINT	0.64	0.11	
XPY	REAL	530.3	0.10	True time could range from 200 - 400 μs, depending on the values of the operands.

Table D.3 Execution Times for the GSV Instruction

Object:	Attribute:	True Time (μs):
PROCESSOR	TimeSlice	16.9
PROCESSORDEVICE	DeviceName	55.2
PROCESSORDEVICE	ProductCode	15.4
PROCESSORDEVICE	ProductRev	15.4
PROCESSORDEVICE	SerialNumber	16.3
PROCESSORDEVICE	Status	15.4
PROCESSORDEVICE	Type	15.4
PROCESSORDEVICE	Vendor	15.4
CST	CurrentStatus	14.4
CST	CurrentValue	28.2
DF1	ACTTimeout	16.5
DF1	DiagnosticCounters	67.2
DF1	DuplicationDetect	14.9
DF1	EmbeddedResponseEnable	14.9
DF1	ENQTransmitLimit	14.9
DF1	EOTSuppression	14.9
DF1	ErrorDetection	14.9
DF1	MasterMessageTransmit	14.9
DF1	NAKReceiveLimit	14.9
DF1	NormalPollGroupSize	15.7
DF1	PollingMode	14.9
DF1	ReplyMessageWait	16.5
DF1	StationAddress	15.7
DF1	SlavePollTimeout	16.5

Table D.3 Execution Times for the GSV Instruction (Continued)

Object:	Attribute:	True Time (μs):
DF1	TransmitRetries	14.9
FAULTLOG	MajorEvents	16.7
FAULTLOG	MinorEvents	16.7
FAULTLOG	MajorFaultBits	17.5
FAULTLOG	MinorFaultBits	17.5
MESSAGE	ConnectionPath	53.1
MESSAGE	ConnectionRate	17.4
MESSAGE	MessageType	15.7
MESSAGE	Port	15.7
MESSAGE	TimeoutMultiplier	15.7
MESSAGE	UnconnectedTimeout	17.4
MODULE	EntryStatus	16.7
MODULE	FaultCode	16.7
MODULE	FaultInfo	17.6
MODULE	ForceStatus	144.5
MODULE	Instance	17.8
MODULE	Mode	16.7
MODULE	LEDStatus	17.2
PROGRAM	DisableFlag	16.4
PROGRAM	Instance	17.8
PROGRAM	LastScanTime	17.6
PROGRAM	MajorFaultRecord	59.0
PROGRAM	MaxScanTime	17.6
PROGRAM	MinorFaultRecord	59.0
PROGRAM	SFCRestart	16.7
ROUTINE	Instance	16.8
SERIALPORT	BaudRate	16.7
SERIALPORT	DataBits	15.0
SERIALPORT	Parity	15.0
SERIALPORT	RTSOffDelay	15.8
SERIALPORT	RTSSendDelay	15.8
SERIALPORT	StopBits	15.0
TASK	Instance	17.7
TASK	LastScanTime	17.5
TASK	MaxInterval	21.6
TASK	MaxScanTime	17.4

Table D.3 Execution Times for the GSV Instruction (Continued)

Object:	Attribute:	True Time (μs):
TASK	MinInterval	21.6
TASK	Priority	16.6
TASK	Rate	17.4
TASK	StartTime	21.6
TASK	Watchdog	17.4
WALLCLOCKTIME	CSTOffset	21.2
WALLCLOCKTIME	CurrentValue	37.6
WALLCLOCKTIME	DateTime	59.8

Table D.4 Execution Times for the SSV Instruction

Object:	Attribute:	True Time (μs):
PROCESSOR	TimeSlice	35.9
DF1	PendingACKTimeout	109.4
DF1	PendingDuplicateDetection	108.3
DF1	PendingEmbeddedResponseEnable	108.7
DF1	PendingENQTransmitLimit	108.3
DF1	PendingEOTSuppression	108.2
DF1	PendingErrorDetection	108.9
DF1	PendingNormalPollGroupSize	108.9
DF1	PendingMasterMessageTransmit	108.7
DF1	PendingNAKReceiveLimit	108.3
DF1	PendingPollingMode	108.7
DF1	PendingReplyMessageWait	109.4
DF1	PendingStationAddress	109.1
DF1	PendingSlavePollTimeout	109.4
DF1	PendingTransmitRetries	108.3
FAULTLOG	MajorEvents	17.0
FAULTLOG	MinorEvents	17.0
FAULTLOG	MajorFaultBits	17.4
FAULTLOG	MinorFaultBits	17.4
MESSAGE	ConnectionPath	36.9
MESSAGE	ConnectionRate	32.3
MESSAGE	MessageType	69.9
MESSAGE	Port	31.6

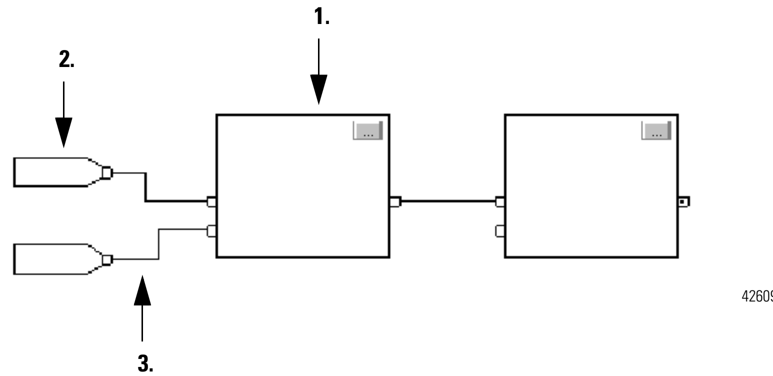
Table D.4 Execution Times for the SSV Instruction (Continued)

Object:	Attribute:	True Time (μs):
MESSAGE	TimeoutMultiplier	31.7
MESSAGE	UnconnectedTimeout	32.3
MODULE	Mode	28.4
PROGRAM	DisableFlag	37.3
PROGRAM	LastScanTime	17.5
PROGRAM	MajorFaultRecord	48.8
PROGRAM	MaxScanTime	17.5
PROGRAM	MinorFaultRecord	48.8
PROGRAM	SFCRestart	33.0
SERIALPORT	PendingBaudRate	108.5
SERIALPORT	PendingDataBits	106.7
SERIALPORT	PendingParity	106.6
SERIALPORT	PendingRTSOffDelay	106.8
SERIALPORT	PendingRTSSendDelay	107.4
SERIALPORT	PendingStopBits	106.7
TASK	LastScanTime	17.7
TASK	MaxInterval	24.6
TASK	MaxScanTime	17.7
TASK	MinInterval	24.6
TASK	StartTime	24.6
TASK	Watchdog	33.8
WALLCLOCKTIME	CSTOffset	2149.4
WALLCLOCKTIME	CurrentValue	70.5
WALLCLOCKTIME	DateTime	120.0

Table D.5 Array Index Times

For a tag in the subscript of a:	Add (μs):
one dimension array	1.5
two dimension array	25.5
three dimension array	30.3

Estimate Execution Time of a Function Block Routine



1. For each function block instruction, add the time from Table D.6 on page D-13.

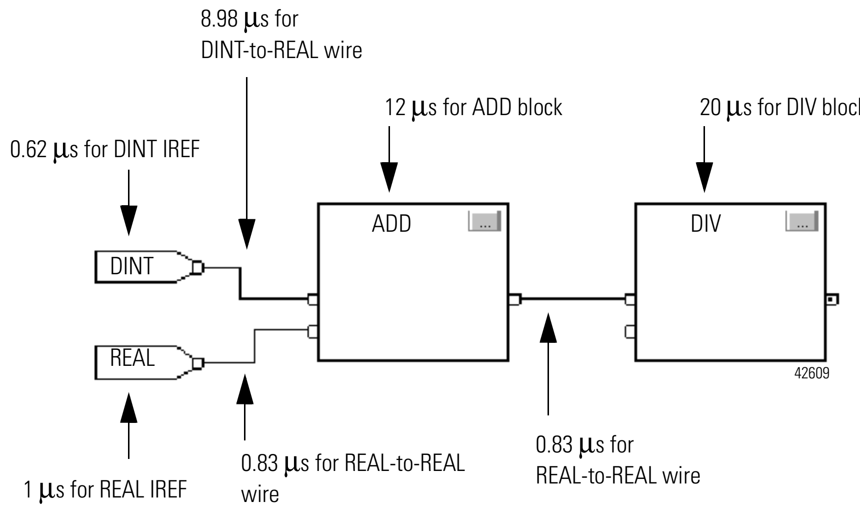
2. For each IREF and OREF, add the following time. (Skip any immediate values.)

For an:	With a data type of:	Add (μs):
IREF	BOOL	0.60
	DINT	0.66
	REAL	1.05
OREF	BOOL	0.17
	DINT	0.17
	REAL	0.17

3. For each wire, add the following time:

If the wire connects a:	Add (μs):
BOOL-to-BOOL	0.43
DINT-to-DINT	0.48
DINT-to-REAL	9.54
REAL-to-DINT	14.08
REAL-to-REAL	0.88

EXAMPLE Estimate the execution time of a function block routine:



For each function block instruction, add the following time:

Table D.6 Execution Times for Function Block Instructions

Instruction:	Time (µs):	Notes:
ABS	2	
ACS	290	
ADD	12	
ALM	98	
AND	1	
ASN	280	
ATN	239	
BAND	10	
BNOT	9	
BOR	12	
BTDT	14	
BXOR	10	
COS	237	
CTUD	16	
D2SD	75	
D3SD	91	
DEDT	102	
DEG	21	

Table D.6 Execution Times for Function Block Instructions (Continued)

Instruction:	Time (μs):	Notes:
DERV	91	
DFF	14	
DIV	21	
EQU	2	
ESEL-Average Sel.	87	
ESEL-High Select	67	
ESEL-Low Select	76	
ESEL-Manual	34	
ESEL-Median Sel.	124	
FGEN	133	
FRD	10	
GEQ	2	
GRT	2	
HLL	26	
HPF	249	
INTG	103	
JKFF	14	
LDL2	223	
LDLG	173	
LEQ	2	
LES	2	
LIM	5	
LN	206	
LOG	206	
LPF	252	
MAVE (uniform)	$68 + (x * 17)$	x = number of samples
MAVE (weighted)	$40 + (x * 11)$	x = number of samples
MAXC	22	
MEQ	2	
MINC	24	
MOD	69	
MSTD	$165 + (x * 51)$	x = number of samples
MUL	20	
MUX	21	
MVMT	13	
NEG	2	

Table D.6 Execution Times for Function Block Instructions (Continued)

Instruction:	Time (μs):	Notes:
NEQ	2	
NOT	1	
NTCH	310	
OR	1	
OSFI	12	
OSRI	13	
PI	141	
PIDE	511	
PMUL	137	
POSP	119	
RAD	21	
RESD	12	
RLIM	94	
RMPS	153	
RTOR	40	
SCL	58	
SCRV	265	
SEL	15	
SETD	11	
SIN	244	
SNEG	18	
SOC	180	
SQR	41	
SRTP	148	
SSUM	$35 + (x * 31)$	x = number of inputs
SUB	12	
TAN	308	
TOD	17	
TOFR	34	
TONR	39	
TOT	115	
TRN	15	
UPDN	25	
XOR	2	
XPY	426	

Notes:

Dynamic Memory Allocation in CompactLogix Controllers

Certain operations cause the controller to dynamically allocate and de-allocate user-available memory, affecting the space available for program logic.

The CompactLogix controller dynamically allocates memory for the following:

- Trend Objects
- Trend Drivers
- Connections

Operations that trigger dynamic memory allocation are:

- Messages
- RSLinx Tag Optimization
- Trends
- DDE/OPC Topics

Although messages are the most likely to cause dynamic memory allocation in CompactLogix controllers, all four operations are discussed in the following sections, along with general guidelines for estimating the amount of memory allocated.

Messages

Messages can come in and go out of the backplane, or come in and go out of the serial port.

Type		Connection Established	Dynamic Memory Allocated
Backplane	Incoming	The message is connected (connection established)	1200 bytes
		The message is unconnected (no connection established)	0 bytes
	Outgoing	All outgoing messages whether connected or unconnnected	1200 bytes
Serial Port	Incoming	All incoming messages whether connected or unconnected	1200 bytes
	Outgoing	All outgoing messages whether connected or unconnected	1200 bytes

RSLinx Tag Optimization

Tag optimization creates three items which allocate memory, a trend object, a trend driver, and a connection.

Item	Description	Memory Allocated
Trend Object	Created in the controller to group the requested tags	80 bytes
Trend Driver	Created to communicate to the trend object	36 bytes/single point (some economy for multiple points in a driver)
Connection	Created between the controller and RSLinx	1200 bytes

EXAMPLE

To monitor 100 points:

100 points x 36 bytes = 3600 bytes (Trend Driver)

3600 (Trend Driver) + 80 (Trend Object) + 1200 (Connection)

= approximately 4000 bytes[1]

(1) In general, we estimate that one tag takes about 40 bytes of memory.

Trends

Each trend created in a controller creates a trend object and allocates a buffer for logging as shown below.

Item	Memory Allocated
Trend Object	80 bytes
Log Buffer	4000 bytes

DDE/OPC Topics

A DDE/OPC Topic uses connections based on the following three variables:

- the number of “Maximum Messaging Connections per PLC” configured in RSLinx
- whether the “Use Connections for Writes to ControlLogix processor” is checked
- the number of connections needed to optimize throughput

IMPORTANT These variables are per path. For example, if you set up two different DDE/OPC topics, with different paths to the same controller, the variables limit the connections for each path. Therefore, if you have a limit of 5 connections, it is possible to have 10 connections, with 5 for each path.

Maximum Messaging Connections per PLC

This variable is configured in RSLinx under the “Communications” menu item “Configure CIP Options”. This number limits the number of read connections made to Logix controllers from the particular workstation.

Checking "Use Connections for Writes to ControlLogix Processor"

This variable is configured in RSLinx under the "Communications" menu item "Configure CIP Options". This check box indicates if you want RSLinx to open up additional connections for writing data to a Logix controller.

NOTE	There is no way to limit the number of write connections, once this box is checked.

Number of Connections Needed to Optimize Throughput

RSLinx only opens the number of connections required to optimize throughput. For example, if you have 1 tag on scan, but have configured RSLinx to allow five connections as the maximum number of connections, RSLinx only opens one connection for the tag. Conversely, if you have thousands of tags on scan and limit the maximum number of CIP connections to five, that is the maximum number of connections that RSLinx establishes to the CompactLogix controller. RSLinx then funnels all of the tags through those five available connections.

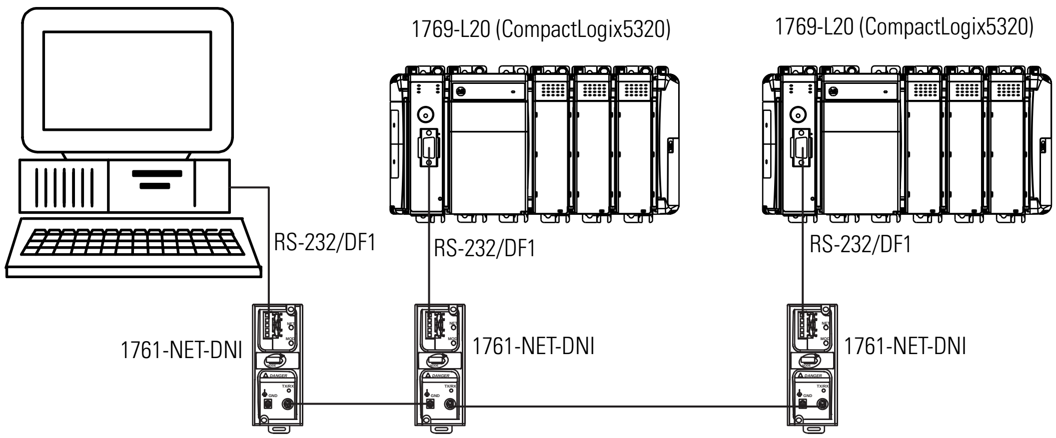
Viewing the Number of Open Connections

You can see how many connections are made from your workstation to the CompactLogix controller in RSLinx by selecting "CIP Diagnostics" from the "Connections" menu. The Dispatching tab contains various CIP information, including the number of connections open to the CompactLogix controller.

Communications on DeviceNet

Using the Serial Port to Connect to a DeviceNet Network

You can use the serial port of the CompactLogix5320 processor to connect to a DeviceNet network using the DeviceNet Interface (DNI), catalog number 1761-NET-DNI. For additional information on using the DNI, refer to the *DeviceNet Interface User Manual*, publication 1761-6.5.



Connecting CompactLogix5320 controllers on DeviceNet requires one 1761-NET-DNI per CompactLogix5320 controller. The DNI converts RS-232 hardware connections and full duplex DF1 protocol to DeviceNet. A computer can also be connected to the DeviceNet network with another 1761-NET-DNI. The Full Duplex DF1 communication driver in RSLinx can be used to allow RSLogix 5000 programming software to upload/download and monitor programs in the CompactLogix controller over the DeviceNet network.

The DNI must be commissioned on the DeviceNet network via the RSNetworx for DeviceNet software or the DNI Configuration Utility, version 2.001.

NOTE The DNI Configuration Utility, a free tool for commissioning and configuring the DNI, is available for download at **www.ab.com**.

Once this is accomplished and all devices on the DeviceNet network have unique node addresses, the CompactLogix5320 controllers can begin exchanging data using MSG Instructions. The Destination Node Address in the DF1 messages sent by the CompactLogix5320 controllers is used by the DNI to route the message to the proper device on DeviceNet.

The following application example shows how to commission the DNI modules on DeviceNet with RSNetworx for DeviceNet software and how to send messages between the CompactLogix5320 controller's serial channels. Refer to the *DeviceNet Interface User Manual*, publication 1761-6.5 for information on the DNI Configuration Utility.

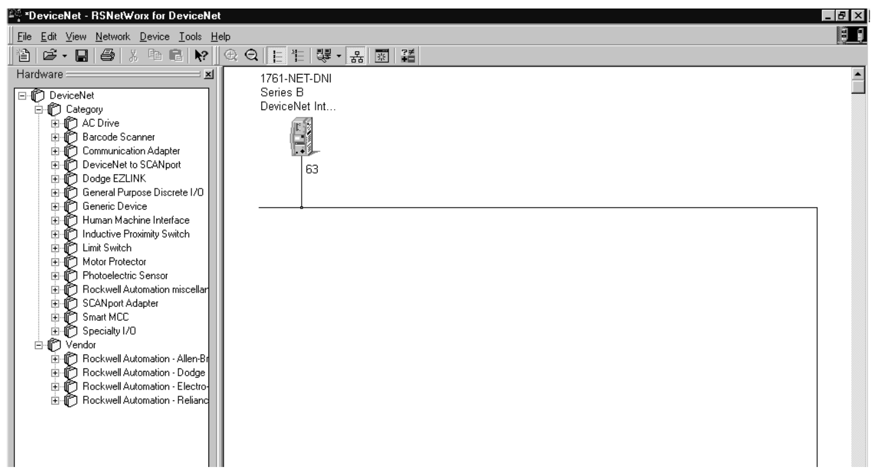
NOTE DeviceNet may be preferred over DH-485 because DeviceNet supports data rates up to 500K bits/second and up to 64 nodes, while DH-485 supports data rates up to 19.2 K bits/second and 32 nodes maximum.

Commissioning the 1761-NET-DNI Modules on DeviceNet

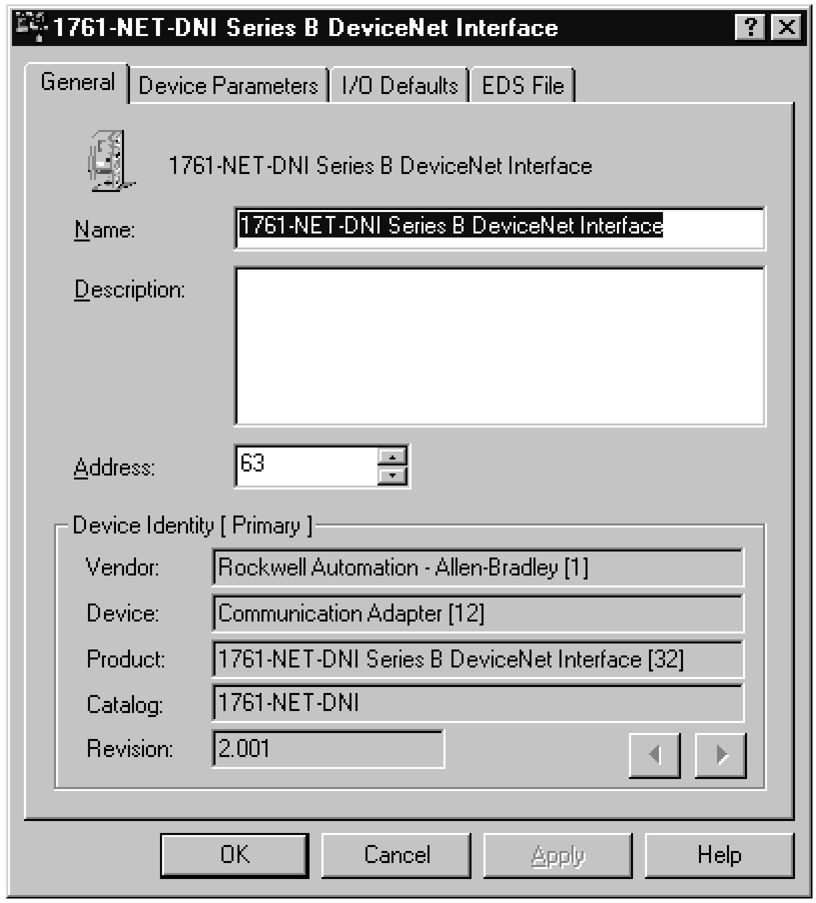
First, commission each DNI module on the DeviceNet network with RSNetworx for DeviceNet. Commissioning is done to assign node addresses to the DNI modules. Each device on the network must have a unique node address. The DNI then routes DF1 messages from each CompactLogix5320 controller to the other CompactLogix5320 controller via DeviceNet and the other DNI module. In this example, after commissioning the DNI modules on the DeviceNet network, we use peer-to-peer messages to send data between CompactLogix5320 controllers.

For a complete description of 1769-NET-DNI features and functionality, refer to the *DeviceNet Interface User Manual*, publication 1761-6.5.

1. Start RSNetworx for DeviceNet by double-clicking its icon. Then connect one of your 1761-NET-DNI modules to the DeviceNet network with your PC running RSNetworx for DeviceNet. When power is first applied to a DNI, it powers up at node 63. In RSNetworx, click on the online icon or click on the “Network” pull-down menu and select Online. The following screen appears:

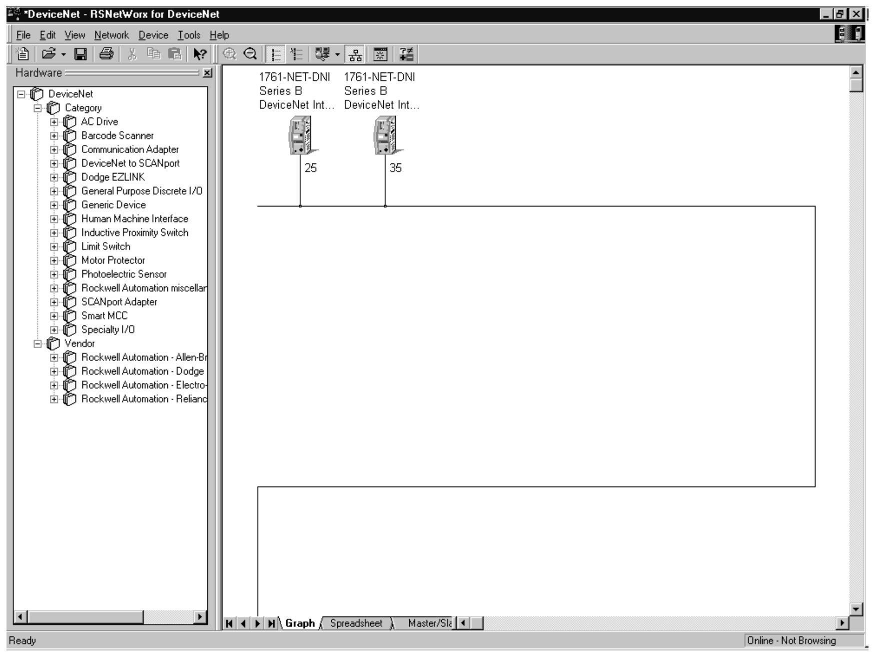


2. The first DNI module appeared as node 63. To change it to any other unique node address, right click on the DNI module and select Properties. The following screen appears:



3. Enter a new node address into the Address field. For this example, addresses 25 and 35 are used for the two DNI module's DeviceNet node addresses. Modify this DNI module's node address to 25. Click Apply and OK.

4. Next, connect the second DNI module to the DeviceNet network. Click the Network pull down menu and select “Single Pass Browse”. The second DNI module should appear at node 63 if the unit is new. Modify its DeviceNet node address as we did with the first DNI module. The two DNI modules should now be at nodes 25 and 35 and the RSNetworx online screen should look like the following:



You have completed the commissioning of your DNI modules with addresses 25 and 35. You can go offline and exit the RSNetworx for DeviceNet software.

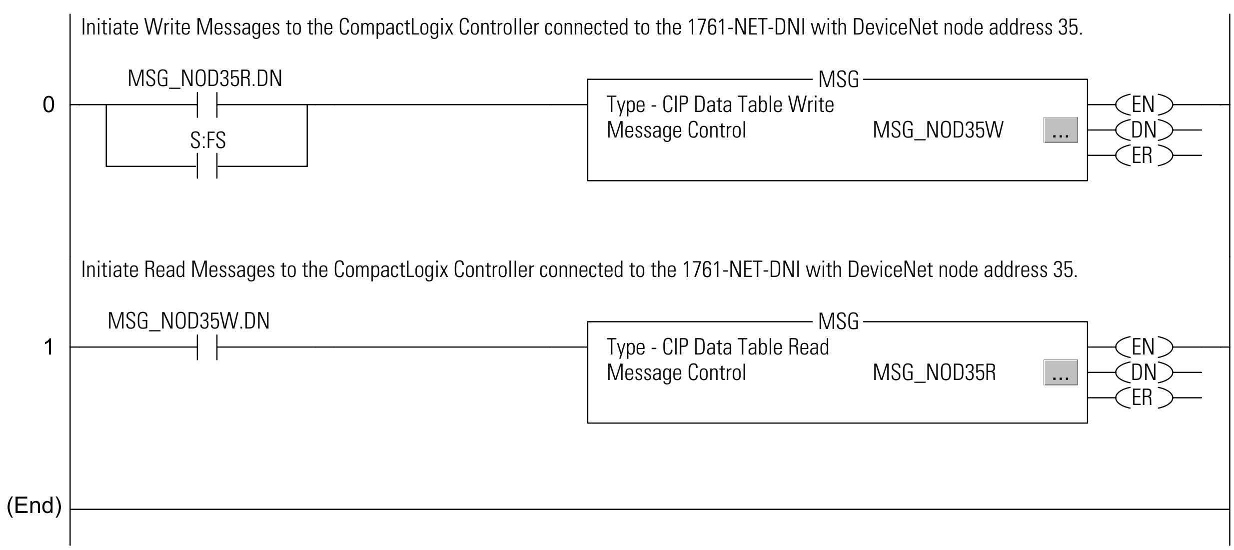
Initializing Messages Between the CompactLogix5320 Controllers on DeviceNet

Now that the DNI modules are configured with unique DeviceNet node addresses, we can connect the two CompactLogix5320 controllers to them and develop a short ladder program to read and write data between them with MSG Instructions.

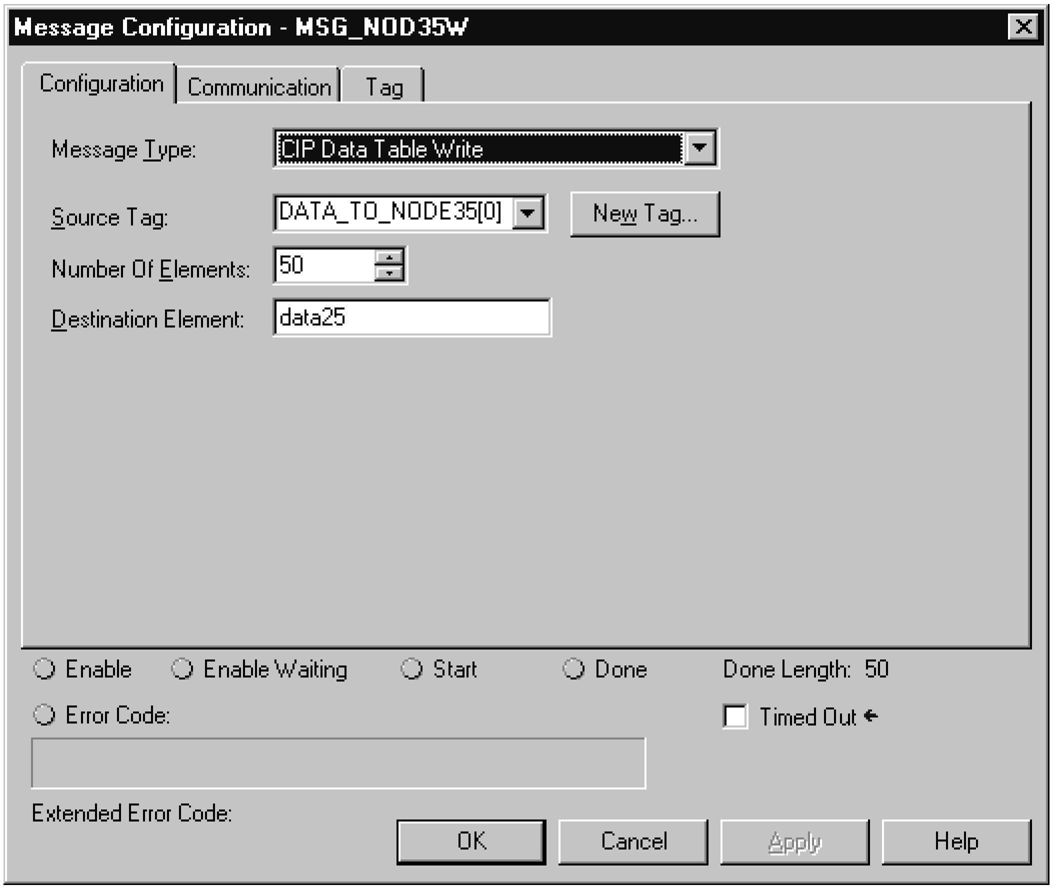
1. Connect the serial channel of each CompactLogix5320 controller to the round mini-din channel on each respective DNI module with a 1761-CBL-PMO2, series B RS-232 cable.

   Assuming that the DNI modules are connected together on the DeviceNet network, any Message sent by either CompactLogix5320 controller with a destination node address of the other controller's DNI module is automatically routed to the other CompactLogix5320 controller. For example, the CompactLogix5320 controller connected to the DNI module with DeviceNet address 25 must send messages to destination node 35. The message is then routed to the controller connected to the DNI module with DeviceNet node address 35.

2. Start RSLogix 5000 and begin a new project for the CompactLogix5320 controller connected to DNI module node 25. Add two ladder rungs, each containing MSG Instructions; one to Write data to the controller and one to Read data from the other controller. For this example, the MSG instructions are executed alternately, initiated at power-up or going to Run mode with the first scan bit (S:FS). This allows each MSG Instruction to re-execute each time it completes. These rungs look like the following:

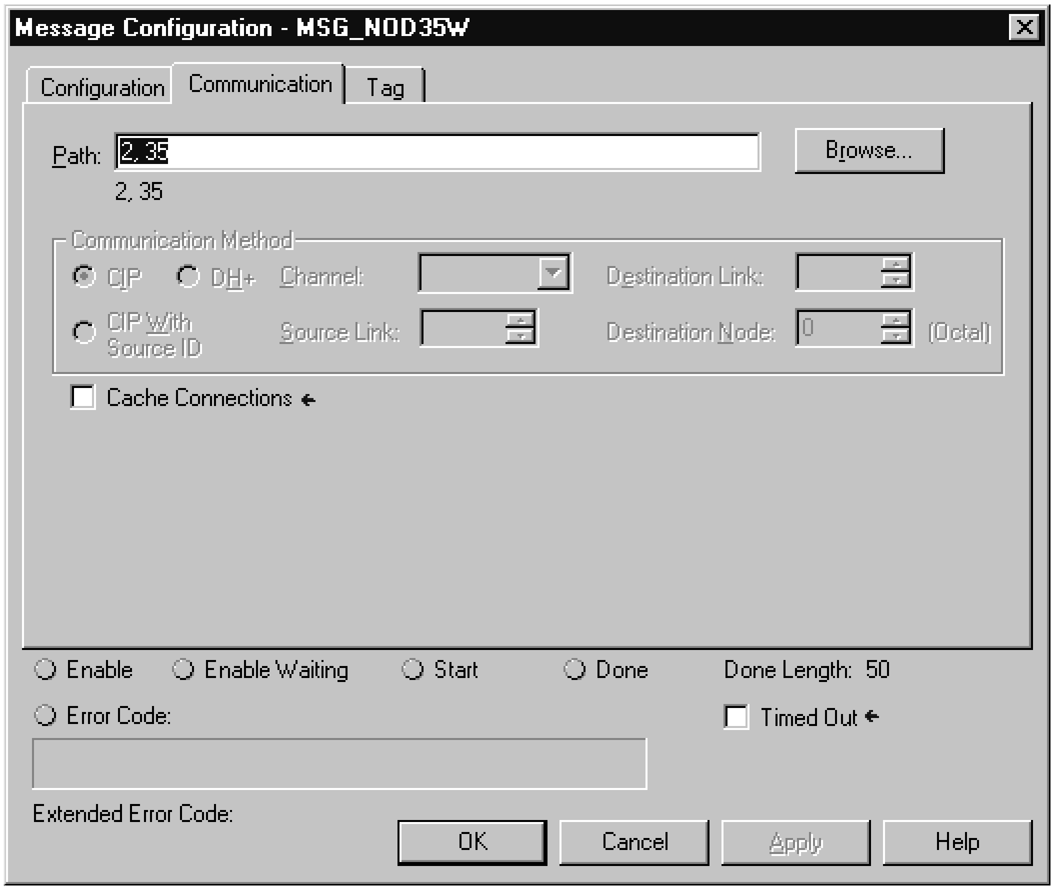


3. The Configuration Tab for the MSG Write Instruction to write fifty 16-bit signed integer data words to the other controller, looks like the following:



A CIP Data Table Write Message Type was used and a 50 integer array tag was created to store the data to write to the other CompactLogix5320 controller. In the controller connected to the DNI node 25, the source tag name for the above Message Instruction is “DATA_TO_NODE35”. This indicates that this data is to be sent to the controller connected to DNI node 35. The Destination Element used in this example is a tag name in the CompactLogix5320 controller connected to DNI node 35. For this example, this tag must be an array of at least 50 integer words in length. It is the destination where the data from this MSG Instruction will be sent.

4. The Communication tab for the MSG Instruction looks like the following:



The Path parameter is the only area on this screen that needs to be modified. The Path shown above (2, 35) routes the Message out port 2 (serial port) of the local CompactLogix5320 controller and to a destination node address 35. This is the DeviceNet node address of the DNI to which the remote CompactLogix5320 controller is connected. The Cache Connections selection is not valid for serial port communications.

The Tag tab merely shows the tag name for the MSG control block. Click Apply and OK to save your changes.

5. The Read message in rung 1 of the ladder logic on page F-6 is very similar to the Write message. The difference is that for the Read message, the Message Type is “CIP Data Table Read”, the Source is “data 25” and the Destination is a tag in the sending controller called “DATA_FROM_NODE35”, for this example. The “data25” tag, located in the CompactLogix5320 controller connected to DNI node 35, is the data table tag location to which both messages in the CompactLogix5320 controller connected to DNI node 25 read and write data.

6. Before saving the project, we must be sure that the port parameters of the serial port match those of the DNI module’s serial port. The default RS-232/Full-duplex DF1 port parameters for the DNI modules and the CompactLogix5320 controllers are identical, except for the error checking method. The DNI module uses the CRC check by default, while the CompactLogix5320 controller’s serial port defaults to BCC. For this example, we change this parameter in the CompactLogix5320 controller.

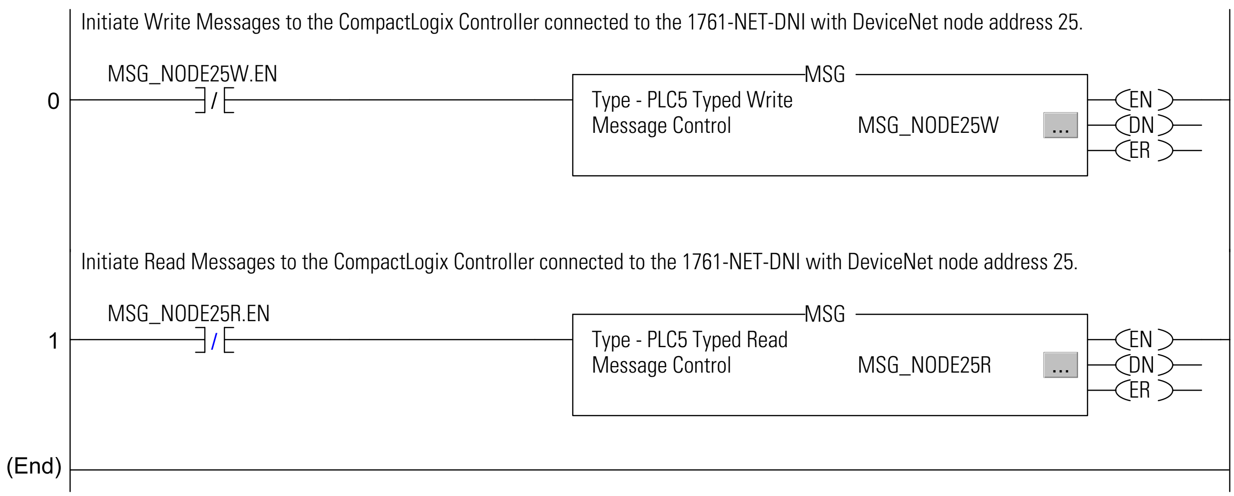
   To accomplish this in each CompactLogix5320 controller project, right-click on the controller name (the upper-most parameter in the Controller Organizer) and select properties. Click on the System Protocol tab and select CRC for the Error Detection. Verify that the DF1 Point-to-Point protocol has been selected. Click on the Serial Port tab to verify that the following default parameters are configured for the serial port. (These parameters match the default parameters of the DNI module’s serial port.)

Parameter	Setting
Mode	System
Baud Rate	19200
Data Bits	8
Parity	None
Stop Bits	1
Control Lines	No Handshaking

   Click APPLY, then OK to save your port configuration parameters.

7. If you wish to have the controller connected to DNI node 35, send read and write messages to the other CompactLogix5320 controller. Add the same two message rungs to it and be sure to create all necessary tags. For this example, a tag named “data35” was created in the CompactLogix5320 connected to DNI node 25. The “data35” tag is an array of 50 integer words so the other CompactLogix5320 can read and write to it.

   The ladder program in the CompactLogix5320 controller connected to DNI node 35 is as follows. The MSG control tags were named to reflect their destination.



8. Download the two RSLogix 5000 project files to their respective CompactLogix5320 controllers and place the controllers into the RUN mode. Each controller will begin reading and writing data to the other controller via their serial ports, the DNI modules and the DeviceNet network.

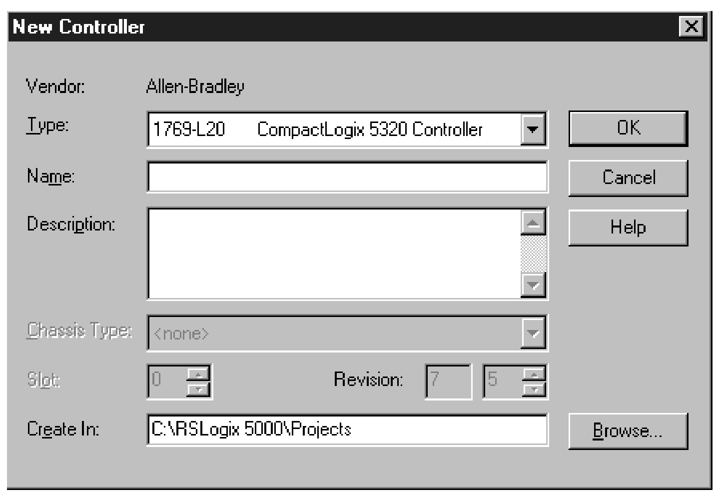
Configuring I/O with the 1769 Generic Profile

Using This Appendix

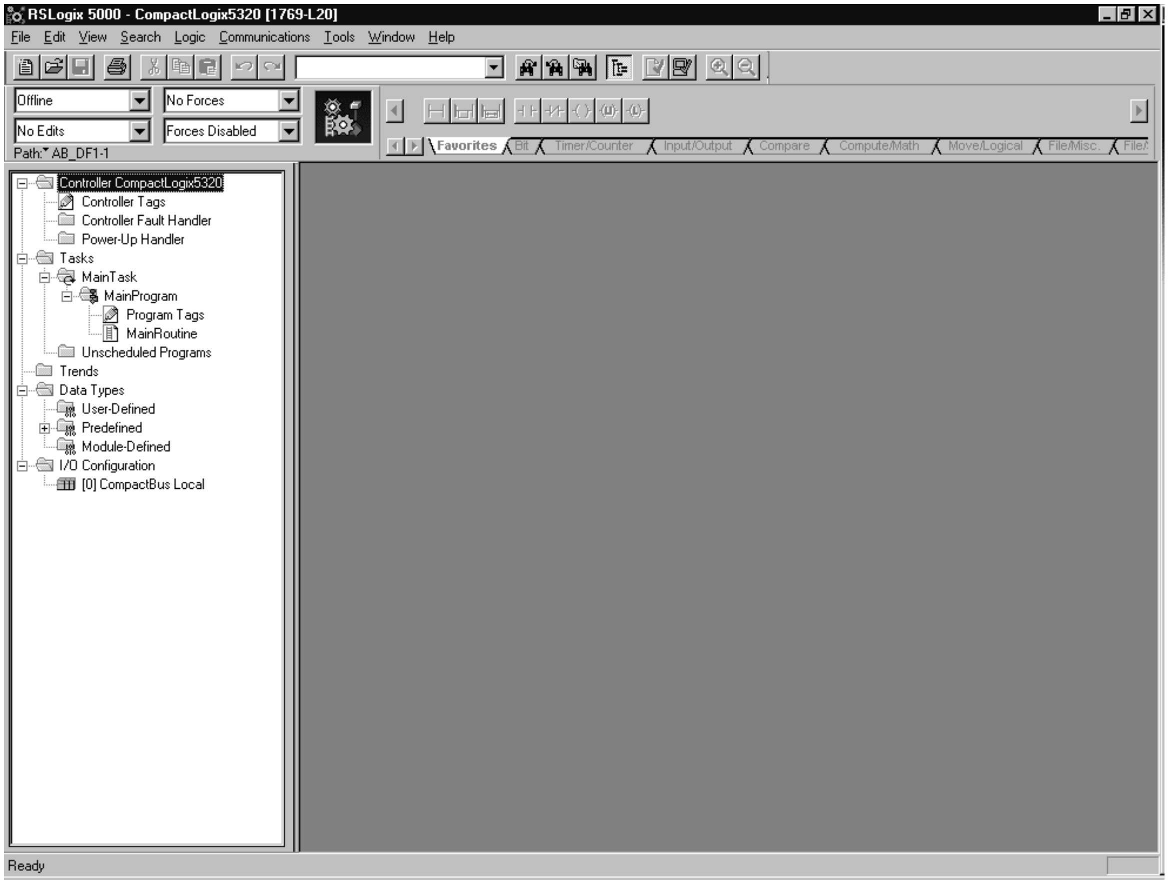
The following generic profile is used only when a specific 1769 I/O module profile is not available in RSLogix 5000 Programming Software. The initial release of the CompactLogix5320 controller includes only the Generic Profile, with individual I/O module profiles to follow.

Creating a Generic Profile

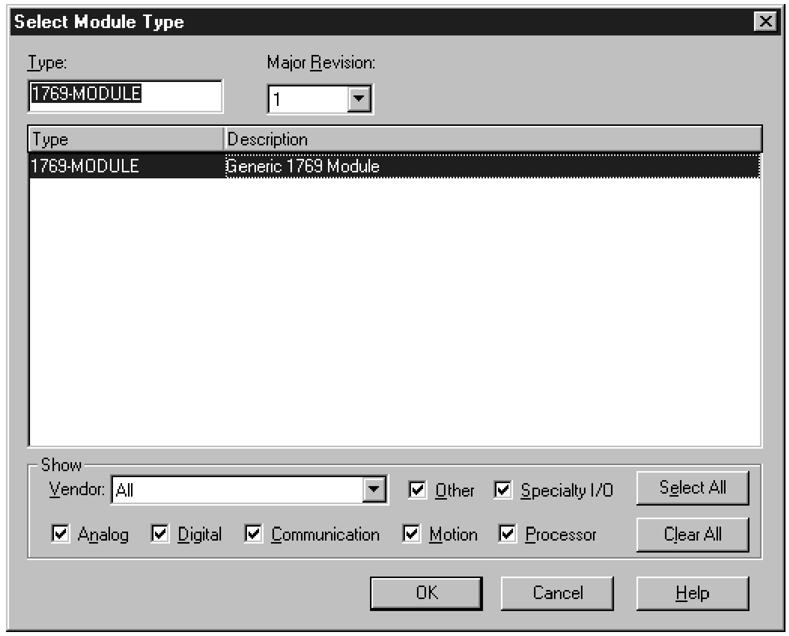
To configure a 1769 I/O module for a CompactLogix Controller in RSLogix 5000 using the Generic Profile, we must first begin a new project in RSLogix 5000. Click on the new project icon or on the File pull down menu and select New. The following screen appears:



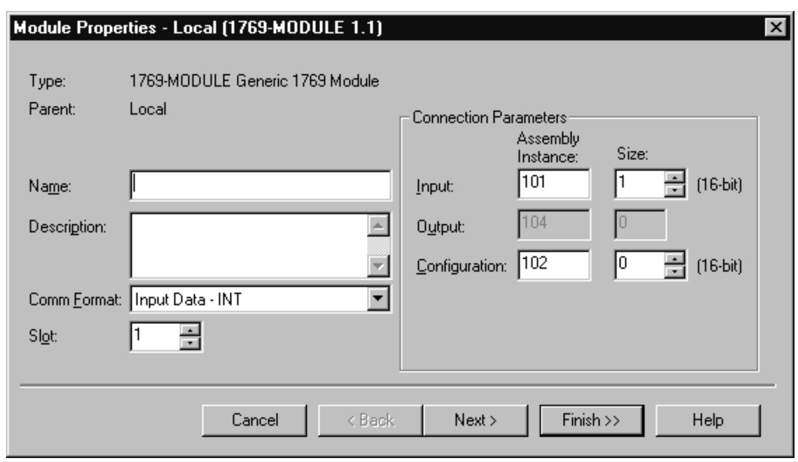
Choose your controller type and enter a name for your project, then click OK. The following main RSLogix 5000 screen appears:



The last entry in the Controller Organizer on the left of the screen shown above is a line labeled “[0] CompactBus Local”. Right click on this line and select New Module. The following screen appears:



This screen is used to narrow your search for I/O modules to configure into your system. With the initial release of the CompactLogix5320 controller, this screen only includes the Generic 1769 Module. Click the OK button and the following default Generic Profile screen appears:



This is the default Generic Profile screen. First, fill in a name for your module. For this example, “OV16” is used for a 1769-OV16 module. This helps to easily identify the module type configured on your local CompactBus. The Description field is optional and may be used to provide more details concerning this I/O module in your application.

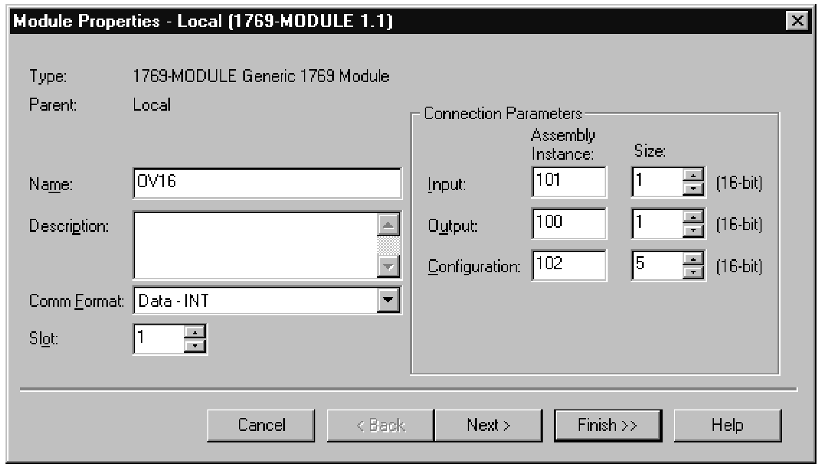
The next parameter is the Comm Format. Click the down arrow for this parameter to reveal the choices. For all 1769 analog output modules, discrete output modules, and discrete combination modules, "Data – INT" is used. "Input Data – INT" is used for all 1769 analog input and discrete input modules. In this example, we create a Generic Profile for the 1769-OV16. The Comm Format is Data – INT.

Next, you must select a slot number. It begins with the first available slot number, 1, and increments automatically for each subsequent Generic Profile you configure. For this example, the 1769-OV16 output module will be located in slot 1.

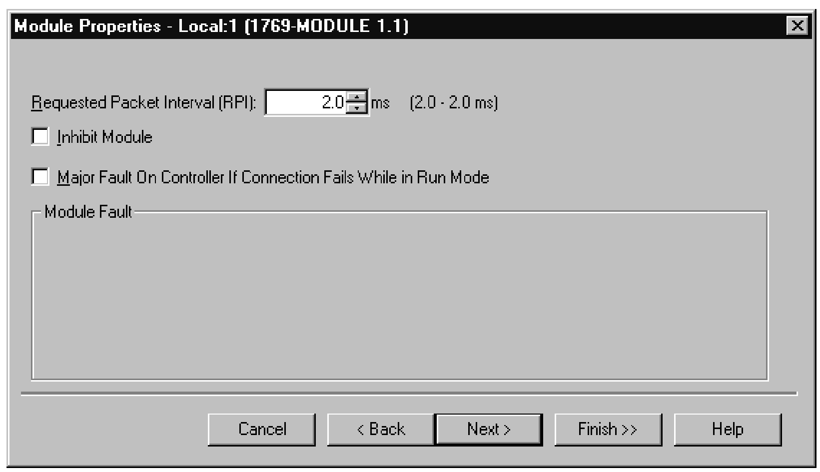
The Comm Format, Assembly Instance and Size values for all 1769 I/O modules are listed in the following table:

1769 I/O Modules	Comm Format	Parameter	Assembly Instance	Size (16-bit)
OA8, OW8, OB16, OV16, OW8I	Data - INT	Input Output Config	101 100 102	1 1 5
IA16, IQ16, IA8, IM12	Input Data – INT	Input Output Config	101 104 102	1 0 0
IQ6XOW4	Data – INT	Input Output Config	101 100 102	2 1 5
IF4	Input Data – INT	Input Output Config	101 104 102	6 0 4
OF2	Data – INT	Input Output Config	101 100 102	4 2 8
IT6, IR6	Input Data – INT	Input Output Config	101 104 102	8 0 8

Note the Comm Format, Assembly Instance numbers and their associated sizes for the 1769-OV16 module type and enter them into the Generic Profile. The Generic Profile for a 1769-OV16 should look like the following:



At this point you may click Finish to complete the configuration of your I/O module. If you click Next, the following screen appears:



The RPI is fixed at 2 msec for the CompactLogix5320 controller. You may choose to inhibit the module, but the “Major Fault On Controller If Connection Fails While In Run Mode” option will have no effect on controller operation. The controller’s response to a connection failure of any I/O module is fixed to always fault the controller. The default is to not inhibit the module. See *Inhibiting the CompactBus* on page 3-6. Please refer to the HELP screens in RSLogix 5000, under Connection Tab Overview for a complete explanation of these features.

Click Finish to complete the configuration of your discrete I/O module.

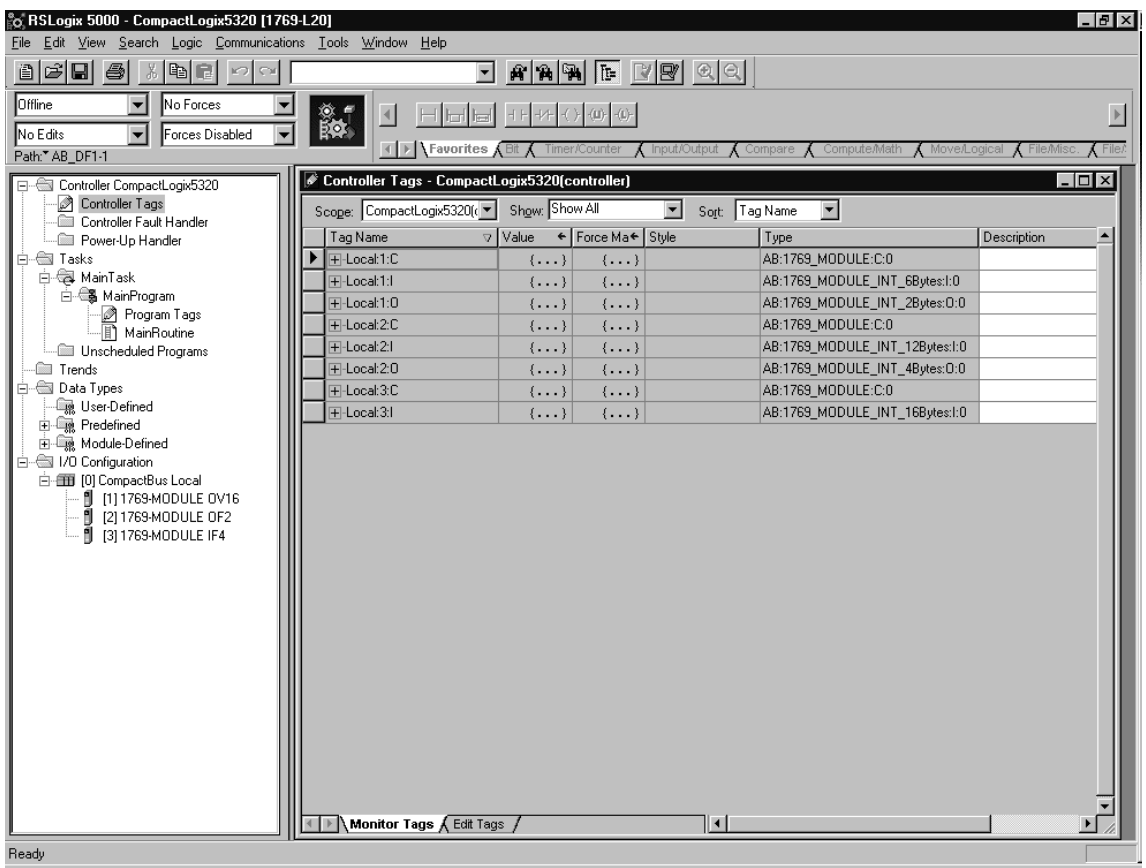
Configure each I/O module in this manner. The CompactLogix5320 controller supports a maximum of 8 I/O modules. The valid slot numbers to select when configuring I/O modules are 1 through 9.

Configuring I/O Modules

Once you have created Generic Profiles for each I/O module in your system, you must enter configuration information into the Tag database that has been automatically created from the Generic Profile information you entered for each of these modules. This configuration information is then downloaded to each module at program download, at power up, and whenever a module is inhibited and then uninhibited.

First enter the Controller Tag database by double-clicking on Controller Tags in the upper portion of the Controller Organizer. In this section, we demonstrate entering configuration data for the 1769-OV16, -OF2 and -IF4 modules. Discrete input modules are not configurable.

For demonstration purposes, Generic Profiles have been created for 1769-OV16, -OF2 and -IF4 modules. As a result, the Controller Tags screen looks like the following:



Tag addresses are automatically created for configured I/O modules. All local I/O addresses are preceded by the word Local. These addresses have the following format:

Input Data: Local:*s*:I

Output Data: Local:*s*:O

Configuration Data: Local:*s*:C

Where *s* is the slot number assigned the I/O modules in the Generic Profiles.

In order to configure an I/O module, you must open the configuration tag for that module by clicking on the plus sign to the left of its configuration tag in the tag data base.

Configuring a 1769 Discrete Output Module

To configure the 1769-OV16 discrete output module in slot 1, click on the plus sign to the left of Local:1:C. Configuration data is entered under the Local:1:C.Data tag. Click the plus sign to the left of Local:1:C.Data to reveal the 5 integer data words where configuration data may be entered for the 1769-OV16 module. The tag addresses for these 5 words are Local:1:C.Data[0] through Local:1:C.Data[4].

Discrete output modules only need to be configured if Hold Last State or User-Defined Safe State features are used. But because the CompactLogix5320 controller does not support either of these features, any data written to these configuration words is disregarded. All outputs will be turned Off when the controller is placed into the Program Mode or when the controller faults.

Discrete output data must be written to tag: Local:1:O.Data[0] and an echo of the output data can be found in tag: Local:1:I.Data[0].

Configuring a 1769-OF2 Analog Output Module

To configure the 1769-OF2 module in slot 2, click on the plus sign to the left of Local:2:C. Configuration data is entered under the Local:2:C.Data tag. Click the plus sign to the left of Local:2:C.Data to reveal the 8 integer data words where configuration data may be entered for the 1769-OF2 module. The tag addresses for these 8 words are Local:2:C.Data[0] through Local:2:C.Data[7]. As indicated in the *Compact I/O Analog Module User's Manual*, publication 1769-UM002A-EN-P, only the first 6 words of the configuration file apply. The last 2 words must exist but both should contain a value of 0 decimal.

The 8 configuration words for the 1769 analog output module are defined as follows:

Word	Bit(s)	Definition
0	0	PFE0 – not used with CompactLogix systems
	1	not used – set to 0
	2	PM0 – not used with CompactLogix systems
	3	FM0 – not used with CompactLogix systems
	4 through 7	not used – set to 0
	8 through 11	Output Type/Range Select Channel 0
	12 through 14	Output Data Format Select Channel 0
	15	Channel 0 Enable Bit
1	0	PFE1 – not used with CompactLogix systems
	1	not used – set to 0
	2	PM1 – not used with CompactLogix systems
	3	FM1 – not used with CompactLogix systems
	4 through 7	not used – set to 0
	8 through 11	Output Type/Range Select Channel 1
	12 through 14	Output Data Format Select Channel 1
	15	Channel 1 Enable Bit
2		Fault Value – Channel 0 – not used with CompactLogix systems
3		Program (Idle) Value – Channel 0 – not used with CompactLogix systems
4		Fault Value – Channel 1 – not used with CompactLogix systems
5		Program (Idle) Value – Channel 1 – not used with CompactLogix systems
6		Must be a value of 0 (decimal)
7		Must be a value of 0 (decimal)

The following is a description of each of the configuration parameters shown on page G-8 for the 1769-OF2 analog output module. A more complete description of these configuration parameters may be found in the *Compact I/O Analog Module User's Manual*, publication 1769-UM002A-EN-P.

Words 0 and 1 contain identical functionality. Word 0 is for channel 0 and Word 1 is for channel 1. See below for a description of each of the parameters in Words 0 and 1.

IMPORTANT The CompactLogix5320 controller does not support the User-Defined Safe State feature. Therefore, the Fault Value and Program (Idle) Value configuration words for analog output modules are not used. When the controller enters the Program Mode or faults, all analog outputs go to a value of 0.

PFE0/PFE1: Program To Fault Enable Bit

This selection is not used with CompactLogix5320 systems.

PM0/PM1: Program Mode Selection Bit

This selection is not used with CompactLogix5320 systems.

FM0/FM1: Fault Mode Selection Bit

This selection is not used with CompactLogix5320 systems.

Output Type/Range Select:

This selection allows you to configure each channel individually for various current or voltage ranges, per the table below:

Output Range	Bit 11	Bit 10	Bit 9	Bit 8
-10V dc to +10V dc	0	0	0	0
0-5V dc	0	0	0	1
0-10V dc	0	0	1	0
4-20ma	0	0	1	1
1-5V dc	0	1	0	0
0-20ma	0	1	0	1

Output Data Format Selection

This selection configures each channel to interpret data presented to it by the controller in any of the following formats:

Output Data Format	Bit 14	Bit 13	Bit 12
Raw/Proportional	0	0	0
Engineering Units	0	0	1
Scaled For PID	0	1	0
Percent Range	0	1	1

NOTE If ±10V dc is selected, the Percent Range output data format is invalid and if chosen, results in a configuration error.

Channel Enable Bit

Bit 15 of Word 0 must be set to a 1 to enable channel 0. Bit 15 of Word 1 must be set to a 1 to enable channel 1. Analog output channels are disabled by default.

Fault Value

This selection is not used with CompactLogix5320 systems.

Program/Idle Value

This selection is not used with CompactLogix5320 systems.

Analog output data must then be written to tags Local:2:O.Data[0] and Local:2:O.Data[1] for channels 0 and 1.

Configuring a 1769-IF4 Analog Input Module

To configure the 1769-IF4 Module in slot 3, click on the plus sign to the left of Local:3:C. Configuration data is entered under the Local:3:C.Data tag. Click on the plus sign to the left of Local:3:C.Data to reveal the 4 integer data words where the configuration data may be entered for the 1769-IF4 module. The tag addresses for these 4 words are Local:3:C.Data[0] through Local:3:C.Data[3]. As indicated in the *Compact I/O Analog Module User's Manual*, publication 1769-UM002A-EN-P, the four configuration words apply to the four input channels respectively.

The four configuration words are all identical, except that each applies to a different analog input channel. Word 0 applies to channel 0, word 1 applies to channel 1 and so on.

These configuration words are as follows:

Bit(s)	Description
0 through 3	Input Filter Select
4 through 7	Reserved – set to 0
8 through 11	Input Type/Range Select
12 through 14	Input Data Format
15	Channel Enable Bit

Input Filter

Each channel configuration word allows you to select an Input Filter on a channel basis. The filter frequency affects the noise rejection characteristics. Filters are selected based on a noise versus step response time trade-off. A 50 Hz filter will provide a high level of noise filtering, resulting in a 60 msec step response. In contrast, a 500 Hz filter will provide minimal noise filtering, resulting in a 6 msec step response.

The following table represents the input filter selections for the 1769-IF4:

Input Filter	Bit 3	Bit 2	Bit 1	Bit 0
60 Hz	0	0	0	0
50 Hz	0	0	0	1
Not Used	0	0	1	0
250 Hz	0	0	1	1
500 Hz	0	1	0	0

Refer to the *Compact I/O Analog Module User's Manual*, publication 1769-UM002A-EN-P, for more information on these filter selections.

Input Type/Range

This selection allows you to configure each channel individually for various current or voltage ranges, as shown in the table below:

Input Type	Bit 11	Bit 10	Bit 9	Bit 8
-10 to +10V dc	0	0	0	0
0 to 5V dc	0	0	0	1
0 to 10V dc	0	0	1	0
4 to 20 mA	0	0	1	1
1to 5V dc	0	1	0	0
0 to 20 mA	0	1	0	1

Input Data Format

The following table provides the various data format selections for each of the analog input channels. For additional details concerning the formats shown below, refer to the *Compact I/O Analog Module User's Manual*, publication 1769-UM002A-EN-P.

Input Data Format	Bit 14	Bit 13	Bit 12
Raw/Proportional	0	0	0
Engineering Units	0	0	1
Scaled for PID	0	1	0
Percent Range	0	1	1

NOTE If ±10V dc is selected, the Percent Range output data format is invalid and if chosen, results in a configuration error.

Channel Enable Bit

Bit 15 in each of the four configuration words for the 1769-IF4 module is the Enable bit for each of the four analog channels. By default, all channels are disabled. To enable a channel, enter a 1 in bit 15 in the appropriate configuration word for that channel. Unused channels should be left disabled.

Analog Input data for the four analog input channels can then be found in tags Local:2:I.Data[0] through Local:2:I.Data[3].

Reach us now at www.rockwellautomation.com

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

Americas Headquarters, 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444
European Headquarters SA/NV, avenue Herrmann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40
Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1769-UM007B-EN-P - February 2001 PN 40072-105-01(B)
Supersedes Publication 1769-UM007A-EN-P - January 2001